REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)* 14-09-2005	**2. REPORT TYPE** Final Report	**3. DATES COVERED** *(From – To)* 1 June 2003 - 27-Mar-06

4. TITLE AND SUBTITLE Fast Prediction of Wing Rock Onset Based on Computational Fluid Dynamics	**5a. CONTRACT NUMBER** FA8655-03-1-3044
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S) Dr. Ken Badcock	**5d. PROJECT NUMBER**
	5d. TASK NUMBER
	5e. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

University of Glasgow
James Watt Building
Glasgow G12 8QQ
United Kingdom

8. PERFORMING ORGANIZATION REPORT NUMBER

N/A

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

EOARD
PSC 821 BOX 14
FPO 09421-0014

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

Grant 03-3044

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

This report results from a contract tasking University of Glasgow as follows: The objective is to develop efficient CFD based techniques for the prediction of wing rock instabilities for a delta wing-body. Two approaches in addition to the basic time marching approach will be investigated. First, direct calculation of the Hopf point for the onset of the instability (for a fixed sweep angle the incidence is regarded as the bifurcation parameter) will be developed based on techniques previously developed for aeroelasticity. Secondly, reduced order models will be developed, driven by CFD results, to again predict the instability onset efficiently. Finally, an assessment of the different methods will be carried out to provide guidance for future application in certification by analysis and design. The derived solutions will also be examined to answer some of the current questions as to the physical mechanisms for wing rock.

15. SUBJECT TERMS

EOARD, Vortex flows, External Aerodynamics, Flight Control

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18, NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** SURYA SURAMPUDI
a. REPORT UNCLAS	**b. ABSTRACT** UNCLAS	**c. THIS PAGE** UNCLAS	UL	39	**19b. TELEPHONE NUMBER** *(Include area code)* +44 (0)20 7514 4299

Fast Prediction of Wing Rock Onset Based on Computational Fluid Dynamics: Final Report on Phases 1 and 2

K. Badcock, M.A.Woodgate and M.R.Allan
Computational Fluid Dynamics Laboratory,
Department of Aerospace Engineering,
University of Glasgow, G12 8QQ, U.K.

P.S.Beran
Multidisciplinary Technologies Center,
Air Force Research Laboratory,
Wright Patterson Air Force Base,
Ohio, 45433-7531, USA.

Abstract

Wing rock is an uncommanded rolling motion which has been observed for slender aircraft. The interaction of vortices is at the heart of the phenomena. Computational fluid dynamics can be used to predict wing rock through time accurate simulation. However, a parametric search based on unsteady calculations is computationally costly. Different approaches to the calculation of wing rock are considered in this reportr, including direct stability calculations to determine onset angles, linearised time domain simulation and reduced order modelling using proper orthogonal decomposition. Finally, the formulation of reduced models for damping and LCO prediction is given.

This work has been funded by the European Office of Aerospace Research and Development under contract FA8655-03-1-3044. This report is the final report for phases 1 and 2 of this contract.

1 Introduction

The desire for increased speed and agility has led to aircraft designs with increasing wing sweep and the addition of highly swept leading edge extensions. A common dynamic phenomenon experienced by slender aircraft flying at high angle of attack is known as wing rock. A classic example of slender wing rock occurred on the Handley Page 115 research aircraft which was designed to study the aerodynamic and handling qualities of slender aircraft at low speed [1]. At angles of attack above 20^o the aircraft first experienced wing rock, with the maximum roll amplitude experienced being around 40^o at 30^o angle of attack. This motion was suppressed by reducing the angle of attack or applying an input to the aileron. Aircraft which experience such motions tend to have highly swept surfaces or have long slender forebodies which produce vortical flows. At some critical angle of attack the aircraft can experience a roll oscillation which grows in amplitude until a limit cycle oscillation is reached. A loss in roll damping is usually associated with wing rock. Although wing rock is a complicated motion which involves several degrees of freedom, the primary motion is a rolling motion around the aircraft longitudinal axis. This motivates experimental and computational studies involving only one degree-of-freedom rolling motions to adequately reproduce the dynamics.

Wing rock of slender delta wings has been studied experimentally over the past two to three decades and several review papers have been published within the last decade [2, 3, 4, 5]. It should also be noted that wing rock can occur for nonslender delta wings [6] and even rectangular wings of low aspect ratio (less than 0.5) [7].

Levin and Katz [8] performed an experimental study into wing rock of 76^o and 80^o sweep delta wings with the wing mounted on a sting-balance. It was observed that only the 80^o wing would undergo self-induced roll oscillations for the given experimental conditions (Reynolds number, bearing friction and wing moment of inertia), therefore it was concluded that the wing aspect ratio must be less than 1 for wing rock. The Reynolds number based on the root chord was 5×10^5 and wing rock was observed for an angle of attack of 20^o. The presence of vortex breakdown over the wing was found to limit the amplitude of the LCO. During the free-to-roll motion a loss in the wing average lift was observed relative to the static lift for the same angle of attack. It was also observed that side forces on the model were high, indicating that the presence of wing rock will have a strong influence on free flight models. Increasing the wind tunnel speed increased the amplitude of the LCO but the reduced frequency of oscillations remains almost unchanged.

Katz and Levin [9] performed an experimental study of a delta wing / canard configuration. The canard has an aspect ratio of 0.7 with the wing having an aspect ratio of 1. The Reynolds number based on the wing's root chord was 3×10^6 and the mechanical friction was minimal (less than 0.2 g-cm). Unlike the results of Levin and Katz [8] where wing rock was not observed for this identical wing, with the reduced mechanical friction in the experimental setup wing rock occurred. The effect of the canard was observed to increase the effective leading edge sweep of the configuration. As such, with the increased effective sweep, the 75^o wing / canard configuration has an enlarged wing rock envelope. Complex vortex interactions were present (since vortices were produced by the canard and the wing) and nonsymmetric oscillations occurred.

Arena and Nelson [10] undertook a series of comprehensive studies on an 80^o sweep delta wing undergoing wing rock. An air bearing spindle was developed for the free-to-roll tests that allowed an isolation of the applied torques due to the flowfield. This implies that the mechanical friction coefficient is effectively zero. Motion history plots were obtained, as well as static and dynamic flow visualisation of vortex position and breakdown location, static surface flow visualisation, steady and unsteady surface pressure distributions. It was observed that there was a rate dependent hysteresis in vortex location due to a time lag in the motion. This time lag was found to produce a dynamically unstable rolling moment able to sustain the wing rock motion.

Rolling motion about the longitudinal axis of a delta wing has been computed using CFD by several researchers [11, 12, 13, 14]. Recently a comprehensive numerical study of wing rock was conducted by Saad [15] using a three degree-of-freedom flight mechanics model for a generic fighter aircraft configuration (forebody, 65^o leading edge sweep, and vertical fin). Roll, sideslip and vertical degrees of freedom were allowed. Including the sideslip degree of freedom was found to delay onset of wing rock and reduce the wing rock amplitude.

Time domain CFD based predictions are a powerful tool for studying wing rock. However, the computational cost can be high. A way of reducing the cost of computing parametric searches for aeroelastic stability behaviour was proposed by Morton and Beran from the US Air Force Laboratories [16]. Their method uses dynamical systems theory to characterise the nature of the aeroelastic instability, with this additional information concentrating the use of the CFD. In this way the problem of locating a one parameter Hopf bifurcation was reduced from multiple time marching calculations to a single steady state calculation of a modified system. This modified system calculates the value of the parameter for which an eigenvalue of the system Jacobian matrix crosses the imaginary axis. A convection-diffusion problem was used to evaluate the approach [17] and the method was then applied to an aeroelastic system consisting of an aerofoil moving in pitch and plunge [16]. The linear system was solved using a direct method and this motivated the use of an approximate Jacobian matrix to reduce the cost. Some robustness problems were encountered when applying the method, particularly at transonic Mach numbers. A complex variable formulation of the problem was introduced in [18] which resolved some of these problems. An approach considered to reduce the difficulties of applying a direct solver to large linear systems was to use domain decomposition to reduce the size of the system at the expense of an outer iteration over the domains. This was tested on the model problem in references [17] and [18].

The problems introduced by using a direct solver were resolved in [19] where a sparse matrix formulation was used to make feasible the solution of the linear system for much larger grids. The Newton iteration was modified to enhance the efficiency of the scheme following work on approximate Jacobian matrices for CFD only problems [20]. The method was shown to be effective for tracing out flutter boundaries for symmetric aerofoils moving in pitch and plunge, with reductions of two orders of magnitude in the computational time required when compared with time marching. This benefit was demonstrated for a three dimensional problem in reference [21].

This report presents the formulation of both a nonlinear and linearised time domain method, a reduced system through Proper Orthogonal Decomposition (POD), an eigenvalue based method using the inverse power method and finally a direct augmented solver. A comparison is made of these methods for an 80 degree delta wing, with an emphasis on evaluating the computational efficiency. The paper continues with the formulation of a solver for the coupled CFD-roll equations, followed by a set of fast methods, direct stability, linearised time domain and proper orthogonal decomposition, which use the full model to predict wing rock onset. The full model is then evaluated for a test case involving an 80-degree wing, with grid dependency, time accuracy and modelling influence considered. Finally the fast methods are compared. The formulation of reduced models based on the direct solution formulation is given in appendices.

2 Formulation

2.1 Flow Solver

All simulations described in this paper were performed using the University of Glasgow PMB (Parallel Multi-Block) RANS solver. A full discussion of the code is given in reference [20]. PMB uses a cell centred finite volume technique to solve the Euler and Reynolds Averaged Navier-Stokes (RANS) equations. The diffusive terms are discretised using a central differencing scheme, and the convective terms are discretised using Osher's approximate Riemann solver with MUSCL interpolation. Steady flow calculations proceed in two parts, initially running an explicit scheme, then switching to an implicit scheme to obtain quicker convergence. The linear system arising at each implicit step is solved using a Krylov subspace method. The pre-conditioning is based on Block Incomplete Lower-Upper BILU(0) factorisation which is decoupled across blocks. For time-accurate simulations, Jameson's pseudo-time (dual-time stepping) formulation [22] is applied, with the steady state solver used to calculate the flow steady states on each physical time step.

For the RANS simulation the two equation k-ω turbulence model is used for closure. It is well known that most linear two-equation turbulence models over-predict the eddy viscosity within vortex cores, thus causing too much diffusion of vorticity [23]. This weakens the vortices and can eliminate secondary separations, especially at low angles of attack where the vortices are weakest. The modification suggested by Brandsma et al. [24] for the production of turbulent kinetic energy was therefore applied to the standard k-ω model of Wilcox [25] to reduce the eddy-viscosity in vortex cores as

$$P_k = min\{P_k^u, (2.0 + 2.0 min\{0, r-1\})\rho\beta^* k\omega\}. \tag{1}$$

Here P_k^u is the unlimited source term for the production of k, and r is the ratio of the magnitude of the rate-of-strain and vorticity tensors. When turbulent kinetic energy is over predicted in the vortex core, it will be limited to a value relative to the dissipation in that region. This modification was found to improve predictions compared with the standard k-ω turbulence model and is used in the RANS simulations presented.

2.2 Semi Discrete System and Jacobian

The coupled CFD-rolling motion equations are written in semi-discrete form as

$$\frac{d\mathbf{w}}{dt} = \mathbf{R} \tag{2}$$

where

$$\mathbf{w} = \begin{bmatrix} \mathbf{w}_a \\ \mathbf{w}_f \end{bmatrix} \tag{3}$$

and

$$\mathbf{R} = \begin{bmatrix} \mathbf{R}_a \\ \mathbf{R}_f \end{bmatrix} \tag{4}$$

are the state and residual vectors, partitioned into aerodynamic and flight variables. Here $\mathbf{w}_a$ is the vector of CFD solution variables in conserved variables in curvilinear coordinates (i.e. $V_i\rho_i$, $V_i\rho_i u_i$, $V_i\rho_i v_i$, $V_i\rho_i w_i$, $V_i E_i$ where the subscript i indicates the cell number, V_i is the cell volume and ρ, u, v, w and E denote the fluid density, three components of velocity and total energy respectively). The vector $\mathbf{R}_a$ is the CFD residual vector resulting from the composition of the cell fluxes calculated using Osher's method. The residual in each cell depends on the aerodynamic variables in 13 cells (i.e. that cell and 4 adjacent cells in each curvilinear direction). The residual also depends on the grid speeds $\mathbf{x}_t$ and the pitch angle of the model α, which is indicated by writing

$$\mathbf{R}_a = \mathbf{R}_a(\mathbf{w}_a, \mathbf{x}_t, \alpha). \tag{5}$$

The roll equations are

$$\frac{d\mathbf{w}_f}{dt} = \mathbf{R}_f \tag{6}$$

where $\mathbf{w}_f = [\phi_t, \phi]^T$ and $\mathbf{R}_f = [\mu C_{lc} + D\phi_t, \phi_t]^T$. Here ϕ is the roll angle about the body axis, C_{lc} is the rolling moment which has been non-dimensionalised as $C_{lc} = l/\rho_\infty U_\infty^2 c_r^3$

where ρ_∞ and U_∞ are the freestream density and velocity respectively, c_r is the centreline chord, D is the coefficient of friction and $\mu = \rho_\infty c_r^5 / I_{xx}$ where I_{xx} is the rolling moment of inertia. The rolling moment depends on the fluid solution but for a fixed fluid solution (in fact for a fixed fluid pressure) is independent of ϕ and α. Hence, the rolling residual is written as

$$\mathbf{R}_f = \mathbf{R}_f(\mathbf{w}_a, \mathbf{w}_f) \tag{7}$$

with the dependence on $\mathbf{w}_s$ coming only from the second component ϕ. The coefficient of friction D is assumed to be zero for the physical model but is exploited below for Hopf Bifurcation calculations.

The full equations can be written to emphasise the dependencies as

$$\frac{d}{dt}\begin{bmatrix} \mathbf{w}_a \\ \phi_t \\ \phi \end{bmatrix} = \begin{bmatrix} \mathbf{R}_a(\mathbf{w}_a, \phi, \phi_t, \alpha) \\ \mu C_{lc}(\mathbf{w}_a) + D\phi_t \\ \phi_t \end{bmatrix}.$$

The stability of an equilibrium $\mathbf{w} = \mathbf{w}_0$ which satisfies $\mathbf{R}(\mathbf{w}_0) = \mathbf{0}$ is determined by the eigensystem of the Jacobian matrix $A = \partial\mathbf{R}/\partial\mathbf{w}$. The matrix can be written out in block format as

$$A = \begin{bmatrix} \frac{\partial \mathbf{R}_a}{\partial \mathbf{w}_a} & \frac{\partial \mathbf{R}_a}{\partial \phi_t} & \frac{\partial \mathbf{R}_a}{\partial \phi} \\ \mu\frac{\partial C_{lc}}{\partial \mathbf{w}_a} & D & 0 \\ 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} A_{aa} & A_{af} \\ A_{fa} & A_{ff} \end{bmatrix}. \tag{8}$$

The term A_{aa} is a large sparse matrix. For A_{fa} the rolling moment depends linearly on the surface pressures, which in turn depend linearly on the values in the first two internal cells from the boundary. The term A_{af} is derived from noting that if initially $\phi = 0$ and the grid is at $\mathbf{x}_0$, and the body axis points along the x-axis, then the instantaneous grid locations are

$$\mathbf{x} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\phi & -sin\phi \\ 0 & sin\phi & cos\phi \end{bmatrix}. \tag{9}$$

The grid speeds are then

$$\mathbf{x}_t = \phi_t \begin{bmatrix} 0 & 0 & 0 \\ 0 & -sin\phi & -cos\phi \\ 0 & cos\phi & -sin\phi \end{bmatrix}. \tag{10}$$

The terms in A_{af} are then calculated through finite differences by incrementing ϕ or ϕ_t, evaluating $\mathbf{x}$ if ϕ has been incremented or $\mathbf{x}_t$ if ϕ_t has been incremented, updating the body boundary conditions and then recalculating the aerodynamic residual for the finite difference calculation.

2.3 Augmented Solution Based on the Critical Angle

2.3.1 Augmented System

We consider the stability problem when the angle of attack α is considered as a parameter. Note that the angle is used to calculate the wing location, keeping the freestream velocity vector parallel to the x-axis.We are interested in finding the onset angle for the wing rock and assume that stability is lost through a Hopf Bifurcation. In this case the Jacobian matrix A has a pair of purely imaginary eigenvalues at the critical angle.

At this angle we therefore have

$$A(\mathbf{P}_1 + i\mathbf{P}_2) = i\omega(\mathbf{P}_1 + i\mathbf{P}_2) \tag{11}$$

where $\mathbf{P}_1$ and $\mathbf{P}_2$ are the real and imaginary parts of the critical eigenvector and $i\omega$ is the purely imaginary eigenvalue. The eigenvector is normalised against a real vector $\mathbf{S}$ so that

$$\mathbf{S}^T(\mathbf{P}_1 + i\mathbf{P}_2) = 0 + i1. \tag{12}$$

At the bifurcation value for α we therefore have a system of conditions

$$\mathbf{R}_{aug} = \begin{bmatrix} \mathbf{R} \\ A\mathbf{P}_1 - \omega\mathbf{P}_2 \\ A\mathbf{P}_2 + \omega\mathbf{P}_1 \\ \mathbf{S}^T\mathbf{P}_1 \\ \mathbf{S}^T\mathbf{P}_2 - 1 \end{bmatrix} = \mathbf{0}. \tag{13}$$

If there are n degrees of freedom in the semi-discrete system state vector $\mathbf{w}$, then equation (13) has 3n+2 equations. Equation (13) can be solved for the augmented vector of unknowns

$$\mathbf{w}_{aug} = \begin{bmatrix} \mathbf{w} \\ \mathbf{P}_1 \\ \mathbf{P}_2 \\ \omega \\ \alpha \end{bmatrix} \tag{14}$$

where $\mathbf{w}$ is the equilibrium at the bifurcation point, $\mathbf{P}_1$ and $\mathbf{P}_2$ are the real and imaginary parts of the critical eigenvector, ω is the frequency of the instability and α is the Hopf bifurcation angle.

The system of equations is solved using Newton's method. One iteration is written as

$$\begin{bmatrix} A & 0 & 0 & 0 & \frac{\partial \mathbf{R}}{\partial \alpha} \\ A_{\mathbf{w}}\mathbf{P}_1 & A & -I\omega & -\mathbf{P}_2 & A_\alpha\mathbf{P}_1 \\ A_{\mathbf{w}}\mathbf{P}_2 & I\omega & A & \mathbf{P}_1 & A_\alpha\mathbf{P}_2 \\ 0 & \mathbf{S}^T & 0 & 0 & 0 \\ 0 & 0 & \mathbf{S}^T & 0 & 0 \end{bmatrix} (\mathbf{w}_{aug}^{n+1} - \mathbf{w}_{aug}^{n}) = -\mathbf{R}_{aug}^{n} \tag{15}$$

There are three problems with using this method. These are now dealt with in turn.

2.3.2 Forming the Augmented Residual

The right hand side vector includes the terms $A\mathbf{P}_1$ and $A\mathbf{P}_2$. These are formed from Residual evaluations through a matrix free evaluation

$$A\mathbf{P} = \frac{\mathbf{R}(\mathbf{w} + \epsilon\mathbf{P}) - \mathbf{R}(\mathbf{w} - \epsilon\mathbf{P})}{2\epsilon}. \tag{16}$$

This avoids the need to form the Jacobian matrix A explicitly. In fact, the matrix free product is used for the contribution of the term A_{aa}.

2.3.3 Forming the LHS matrix

Approximations to the LHS matrix are possible as long as the iterations converge at a reasonable rate to the correct solution. It has been found that for CFD only and aeroelastic augmented bifurcation calculations that using the Jacobian matrix A_{aa} corresponding to the first order spatial scheme leads overall to a more efficient scheme. When this approximation is made, the resulting Jacobian matrix corresponding to equation (8) is denoted by $\overline{A}$. This is because, although quadratic convergence is lost and so more quasi

Newton steps are needed, the linear system is much easier to solve. The approximate Jacobian involves seven non-zero blocks per cell in the grid. The second Jacobian $A_{\mathbf{w}}$ is difficult to compute efficiently and this term is also neglected in the approximate Jacobian driving the augmented Newton solver. This Jacobian matrix is then written as

$$\overline{\frac{\partial \mathbf{R}_{aug}}{\partial \mathbf{w}_{aug}}} = \begin{bmatrix} \overline{A} & 0 & 0 & 0 & 0 \\ 0 & \overline{A} & -I\omega & -\mathbf{P}_2 & A_\alpha \mathbf{P}_1 \\ 0 & I\omega & \overline{A} & \mathbf{P}_1 & A_\alpha \mathbf{P}_2 \\ 0 & \mathbf{S}^T & 0 & 0 & 0 \\ 0 & 0 & \mathbf{S}^T & 0 & 0 \end{bmatrix}. \tag{17}$$

The terms $A_\alpha \mathbf{P}_1$ and $A_\alpha \mathbf{P}_2$ are formed using a finite difference approximation. The first equation, which calculates the equilibrium solution at the current angle, is decoupled from the critical eigenvalue equations.

2.3.4 Forming the Exact Jacobian Matrix

The key issue is how the convective terms in the Euler equations are treated. All other terms in the coupled Jacobian matrix are treated either analytically or by finite differences as described in previous reports. We will therefore focus on the Jacobian of the fluid residual $\mathbf{R}_a$ with respect to the fluid unknowns $\mathbf{w}_a$.

The residual for one cell in the grid is built up from fluxes. Following the usual approach for Riemann solvers,

$$\mathbf{f}_{i,j} = \mathbf{f}_{i,j}(\mathbf{w}_l, \mathbf{w}_r)$$

where $\mathbf{w}_l = \mathbf{w}_l(\mathbf{w}_{i-2,j}, \mathbf{w}_{i-1,j}, \mathbf{w}_{i,j}, \mathbf{w}_{i+1,j})$ and $\mathbf{w}_r = \mathbf{w}_r(\mathbf{w}_{i-1,j}, \mathbf{w}_{i,j}, \mathbf{w}_{i+1,j}, \mathbf{w}_{i+2,j})$. Here $\mathbf{f}_{i,j}$ is computed using Osher's [26] approximate Riemann solver. The left and right states are computed from the cell values using MUSCL interpolation. For the cell interface we are considering there are four contributions to the Jacobian matrix arising from

$$\frac{\partial \mathbf{f}_{i,j}}{\partial \mathbf{w}_{i-2,j}}$$

$$\frac{\partial \mathbf{f}_{i,j}}{\partial \mathbf{w}_{i-1,j}}$$

$$\frac{\partial \mathbf{f}_{i,j}}{\partial \mathbf{w}_{i,j}}$$

$$\frac{\partial \mathbf{f}_{i,j}}{\partial \mathbf{w}_{i+1,j}}$$

Now, for the approximate Jacobian used to drive convergence of the Newton iterations, the following approximations were made:

$$\frac{\partial \mathbf{f}_{i,j}}{\partial \mathbf{w}_{i-2,j}} \approx \mathbf{0}$$

$$\frac{\partial \mathbf{f}_{i,j}}{\partial \mathbf{w}_{i-1,j}} \approx \frac{\partial \mathbf{f}_{i,j}}{\partial \mathbf{w}_l}$$

$$\frac{\partial \mathbf{f}_{i,j}}{\partial \mathbf{w}_{i,j}} \approx \frac{\partial \mathbf{f}_{i,j}}{\partial \mathbf{w}_r}$$

$$\frac{\partial \mathbf{f}_{i,j}}{\partial \mathbf{w}_{i+1,j}} \approx \mathbf{0}$$

These approximations are exact for a first order spatial discretisation where $\mathbf{w}_l = \mathbf{w}_{i-1,j}$ and $\mathbf{w}_r = \mathbf{w}_{i,j}$. The calculation of the terms

$$\frac{\partial \mathbf{f}_{i,j}}{\partial \mathbf{w}_l}$$

and

$$\frac{\partial \mathbf{f}_{i,j}}{\partial \mathbf{w}_r}$$

is non-trivial but has been coded, tested and used in the Glasgow CFD solver since 1998 [28]. These are exploited to calculate the exact Jacobian terms for the second order spatial discretisation by using the Chain rule

$$\frac{\partial \mathbf{f}_{i,j}}{\partial \mathbf{w}_{i-2,j}} = \frac{\partial \mathbf{f}_{i,j}}{\partial \mathbf{w}_l}\frac{\partial \mathbf{w}_l}{\partial \mathbf{w}_{i-2,j}}$$

$$\frac{\partial \mathbf{f}_{i,j}}{\partial \mathbf{w}_{i-1,j}} = \frac{\partial \mathbf{f}_{i,j}}{\partial \mathbf{w}_l}\frac{\partial \mathbf{w}_l}{\partial \mathbf{w}_{i-1,j}} + \frac{\partial \mathbf{f}_{i,j}}{\partial \mathbf{w}_r}\frac{\partial \mathbf{w}_r}{\partial \mathbf{w}_{i-1,j}}$$

$$\frac{\partial \mathbf{f}_{i,j}}{\partial \mathbf{w}_{i,j}} = \frac{\partial \mathbf{f}_{i,j}}{\partial \mathbf{w}_l}\frac{\partial \mathbf{w}_l}{\partial \mathbf{w}_{i,j}} + \frac{\partial \mathbf{f}_{i,j}}{\partial \mathbf{w}_r}\frac{\partial \mathbf{w}_r}{\partial \mathbf{w}_{i,j}}$$

$$\frac{\partial \mathbf{f}_{i,j}}{\partial \mathbf{w}_{i+1,j}} = \frac{\partial \mathbf{f}_{i,j}}{\partial \mathbf{w}_r}\frac{\partial \mathbf{w}_r}{\partial \mathbf{w}_{i+1,j}}$$

The new coding of the terms arising from the dependence of $\mathbf{w}_l$ and $\mathbf{w}_r$ on the cell values is shown in appendix A.

Some care must be taken at boundaries where halo cells are used to simplify implementation. The halo values are functions of the internal values $\mathbf{w}_{b1} = \mathbf{w}_{b1}(\mathbf{w}_1, \mathbf{w}_2)$ and $\mathbf{w}_{b2} = \mathbf{w}_{b2}(\mathbf{w}_1, \mathbf{w}_2)$ with the exact relationship depending on the type of boundary. Applying the Chain rule,

$$\frac{\partial \mathbf{f}_b}{\partial \mathbf{w}_1} = \frac{\partial \mathbf{f}_b}{\partial \mathbf{w}_l}\frac{\partial \mathbf{w}_l}{\partial \mathbf{w}_1} + \frac{\partial \mathbf{f}_b}{\partial \mathbf{w}_r}\frac{\partial \mathbf{w}_r}{\partial \mathbf{w}_1} + \frac{\partial \mathbf{f}_b}{\partial \mathbf{w}_l}\frac{\partial \mathbf{w}_l}{\partial \mathbf{w}_{b1}}\frac{\partial \mathbf{w}_{b1}}{\partial \mathbf{w}_1} + \frac{\partial \mathbf{f}_b}{\partial \mathbf{w}_r}\frac{\partial \mathbf{w}_r}{\partial \mathbf{w}_{b1}}\frac{\partial \mathbf{w}_{b1}}{\partial \mathbf{w}_1} + \frac{\partial \mathbf{f}_b}{\partial \mathbf{w}_l}\frac{\partial \mathbf{w}_l}{\partial \mathbf{w}_{b2}}\frac{\partial \mathbf{w}_{b2}}{\partial \mathbf{w}_1}$$

and

$$\frac{\partial \mathbf{f}_b}{\partial \mathbf{w}_2} = \frac{\partial \mathbf{f}_b}{\partial \mathbf{w}_l}\frac{\partial \mathbf{w}_l}{\partial \mathbf{w}_2} + \frac{\partial \mathbf{f}_b}{\partial \mathbf{w}_r}\frac{\partial \mathbf{w}_r}{\partial \mathbf{w}_2} + \frac{\partial \mathbf{f}_b}{\partial \mathbf{w}_l}\frac{\partial \mathbf{w}_l}{\partial \mathbf{w}_{b1}}\frac{\partial \mathbf{w}_{b1}}{\partial \mathbf{w}_2} + \frac{\partial \mathbf{f}_b}{\partial \mathbf{w}_l}\frac{\partial \mathbf{w}_l}{\partial \mathbf{w}_{b2}}\frac{\partial \mathbf{w}_{b2}}{\partial \mathbf{w}_2}.$$

The dependence of the halo values on the interior values leads to similar extra terms from the adjacent interfaces to the boundary also.

The second order Jacobians were tested by forming matrix vector products against random vectors and comparing with the results from a matrix free product.

2.3.5 Initial Values for the Eigenvector

The convergence of the Newton iterates requires a good initial guess. We assume that we have an estimate for α and ω. Then a steady state CFD calculation gives $\mathbf{w}_a$, with $\mathbf{w}_s = \mathbf{0}$. Finally, the inverse power method is applied to the approximate Jacobian $\overline{A}$ or the exact Jacobian A, using ω as a shift to give an initial guess for $\mathbf{P}_1$ and $\mathbf{P}_2$.

2.3.6 Augmented Solution Based on the Friction Coefficient

The augmented solver described above suffers from a potential difficulty. The fluid solution is strongly dependent on the angle of attack. This means that decoupling the equilibrium and stability equations is not a good strategy. A structural parameter makes a better bifurcation parameter since the fluid equations, which are far more numerous, are potentially less sensitive to a structural parameter. We therefore consider a second approach to calculating the bifurcation angle.

Let us consider the case where the angle of attack is fixed and the structural friction coefficient D is regarded as the bifurcation parameter. This has the following advantages

- The fluid and structural equilibrium is independent of D and therefore the first component of equation (13) decouples from the other equations. Several of the approximations made in obtaining equation (17) are therefore not necessary.
- The terms A_{aa}, A_{fa} and A_{af} are all independent of D and so can be calculated once and for all at the start of the Newton iterations.
- A consequence of this is that the incomplete L-U decomposition can also be calculated at the start of the Newton iterations and then held fixed. This is likely to be a good approximation since most of the matrix does not change during the Newton iterations.
- Finally, the terms $A_D P_1$ and $A_D P_2$ are, except for one component, equal to zero.

The onset friction coefficient can be computed through the following scheme. First, the fluid steady state is computed. Secondly, the inverse power method is applied as described above to provide initial values for the real and imaginary parts of the eigenvector. Thirdly, the matrices A_{aa}, A_{fa} and A_{af} are computed and fixed for the rest of the computation. The augmented Newton iterations are then driven by

$$\overline{\frac{\partial \mathbf{R}_{aug}}{\partial \mathbf{w}_{aug}}} = \begin{bmatrix} \overline{A} & -I\omega & -\mathbf{P}_2 & A_D\mathbf{P}_1 \\ I\omega & \overline{A} & \mathbf{P}_1 & A_D\mathbf{P}_2 \\ \mathbf{S}^T & 0 & 0 & 0 \\ 0 & \mathbf{S}^T & 0 & 0 \end{bmatrix}. \quad (18)$$

where the solution variables are

$$\mathbf{w}_{aug} = \begin{bmatrix} \mathbf{P}_1 \\ \mathbf{P}_2 \\ \omega \\ D \end{bmatrix}. \quad (19)$$

The LU decomposition of the matrix in equation (18) is calculated at the first iteration and is then fixed thereafter.

Now, the model friction is zero for a free-flying aircraft and so basing the bifurcation calculation on the onset friction is not very useful in itself. However, this calculation can be efficiently made at a number of angles of attack and the value of onset friction which is zero computed. When this happens we have the onset angle for zero model friction.

2.4 Linearised Solver and POD

A simpler version of the time domain solver which is likely to be adequate for the prediction of stability can be obtained by linearising the semi-discrete system about a steady state. This also provides a starting point for developing a reduced order model using proper orthogonal decomposition. The state vector $\mathbf{w}$ is written as $\mathbf{w} = \mathbf{w}_0 + \mathbf{w}'(t)$ where $\mathbf{w}_0$ is an equilibrium value and $\mathbf{w}'(t)$ is the perturbation about this mean. To first order

$$\frac{d\mathbf{w}'}{dt} = \mathbf{R}(\mathbf{w}_0) + \frac{\partial \mathbf{R}}{\partial \mathbf{w}}\mathbf{w}' \tag{20}$$

and since $\mathbf{w}_0$ is an equilibrium, $\mathbf{R}(\mathbf{w}_0) = \mathbf{0}$. Hence,

$$\frac{d\mathbf{w}'}{dt} = A\mathbf{w}' \tag{21}$$

to first order.

Now, given a set of solutions from a time marching calculation, a set of modes can be computed by proper orthogonal decomposition (POD) The computed modes are written in matrix form as

$$\mathbf{\Phi} = [\phi_1|\phi_2|.........|\phi_n]\,. \tag{22}$$

The projection of the solution onto the modes is $\mathbf{w}' = \Phi\hat{\mathbf{w}}$, where $\hat{\mathbf{w}}$ is a vector of modal coordinates. By using the projected solution, equation (21) can be written as

$$\Phi\frac{d\hat{\mathbf{w}}}{dt} = A\Phi\hat{\mathbf{w}} \tag{23}$$

and premultiplying by Φ^T gives

$$\frac{d\hat{\mathbf{w}}}{dt} = \Phi^T A\Phi\hat{\mathbf{w}} = \hat{A}\hat{\mathbf{w}}. \tag{24}$$

The matrix $\hat{A}$ is the modal Jacobian matrix whose components can be formed as

$$\hat{A}_{ij} = \phi_i^T \frac{\mathbf{R}(\mathbf{w}_0 + \epsilon\phi_j) - \mathbf{R}(\mathbf{w}_0 - \epsilon\phi_j)}{2\epsilon}. \tag{25}$$

Once the elements of $\hat{A}$ have been computed, requiring 2 residual evaluations for each mode retained, the stability of the equilibrium can be tested for the α used to train the modes using the small dimension linear system of ODE's in equation (42).

This method requires the calculation of mode shapes using snapshots from time marching calculations. The formulation is based on primitive variables i.e. the mode shapes are calculated in terms of snapshots expressed in terms of ρ, u, v, w and p. An equation for the evolution of the primitive equations can be derived using the chain rule as

$$\frac{\partial \mathbf{w}}{\partial \mathbf{p}}\frac{d\mathbf{p}}{dt} = \mathbf{R}(\mathbf{w}) \tag{26}$$

which gives

$$\frac{d\mathbf{p}}{dt} = \frac{\partial \mathbf{p}}{\partial \mathbf{w}}\mathbf{R}(\mathbf{w}) = \mathbf{R}^*(\mathbf{p}). \tag{27}$$

The solution is then linearised as $\mathbf{p} = \mathbf{p}_0 + \mathbf{p}'(t)$ and then $\mathbf{p}' = \Phi^*\hat{\mathbf{p}}$. Finally, the projected equation is

$$\frac{d\hat{\mathbf{p}}}{dt} = \Phi^{*T} A\Phi^*\hat{\mathbf{p}} = \hat{A}^*\hat{\mathbf{p}} \tag{28}$$

where

$$\hat{A}^*_{ij} = \phi_i^{*T}\frac{\mathbf{R}^*(\mathbf{p}_0 + \epsilon\phi_j^*) - \mathbf{R}^*(\mathbf{p}_0 - \epsilon\phi_j^*)}{2\epsilon}. \tag{29}$$

3 Optimisation of Linear Solver

3.1 Overview

A key part of the computational work for the methods described in previous reports is the solution of a large sparse linear system. The package Aztec [29] was used for this purpose. Aztec is a general sparse matrix package which implements Krylov subspace solvers, several preconditioners and has a parallel option. A general block structure for the matrix can be accomodated.

The key decisions before parallel implementation of the stability prediction methods concern the linear solver. In particular, the preconditioner used is a block incomplete LU decomposition (BILU) [30] which is sequential in nature. Extra approximations are needed to obtain a method which is effective in parallel. Efforts were directed first at examining the sequential performance of the linear solver. Then, a parallel version was considered. This work is described in the current section.

3.2 GCR Algorithm

Eisenstat, Elman and Schultz [27] developed a generalized Conjugate Gradient mehtod that depends only on A rather than $A^T A$ and is called the Generalized Conjugate Residual (GCR) algorithm. Saad and Schultz developed the Generalized Minimal Residual (GMRES) algorithm which is mathematically equivalent to GCR but is less prone to breakdown for certain problems, requires less storage and arithmetic operations. However GCR remains the easier algorithm to implement, especially in parallel. The GCR algorithm is:

```
$r_0 = b - Ax_0$
$p_0 = r_0$
For $j = 0, 1, 2, \dots,$ until convergence. Do:
    $\alpha_j = \frac{\langle r_j, Ap_j \rangle}{\langle Ap_j, Ap_j \rangle}$
    $x_{j+1} = x_j + \alpha_j p_j$                                   (30)
    $r_{j+1} = r_j - \alpha_j Ap_j$
    $\beta_{ij} = -\frac{\langle Ar_{j+1}, Ap_i \rangle}{\langle Ap_i, Ap_i \rangle}$, for $i = 0, 1, 2, \dots, j$
    $p_{j+1} = r_{j+1} + \sum_{i=0}^{j} \beta_{ij} p_i$
Enddo
```

To calculate the β_{ij} the vector Ar_j and the previous Ap_j's are required. The number of matrix vector products per step can be reduced to one if Ap_{j+1} is calculated by

$$Ap_{j+1} = Ar_{j+1} + \sum_{i=0}^{j} \beta_{ij} Ap_i \tag{31}$$

This may not be beneficial if A is sparse and j is large.

A restarted version called GCR(m) is defined so that when the iteration reaches step m all the p_j's and Ap_j's are thrown away. Since GCR and GMRES are closely linked it might be possible to use one of the standard GMRES subbase methods to improve convergence.

3.3 BILU(k)

For the Block Incomplete Lower Upper (BILU) factorization the matrix is partitioned into 5 × 5 matrix blocks associated with each cell in the mesh. The use of this blocking reduces the memory required to store the matrix in a sparse matrix format. For clarity matrix elements refer to these blocks from now on.

Consider a general sparse matrix A whose elements are a_{ij}, $i, j = 1, \ldots, n$. A general Incomplete factorization computes a sparse lower triangular matrix L and a sparse upper triangular matrix U so that the residual matrix $R = LU - A$ satisfies certain contraints, such as having entries in a prescribed pattern. A common constraint consists of taking the zero pattern of the L U factors to be precisely the zero pattern of A. However the accuracy of the ILU(0) incomplete factorization may be insufficient to provide an adequate rate of convergence.

More accurate block Incomplete LU factorisations allowing extra terms to be filled into the factorisation are often more efficient as well as more robust. Consider updating the a_{ij} element in full Gaussian Elimination (GE). The inner loop contains the equation

$$a_{ij} = a_{ij} - a_{ik}a_{kj}. \tag{32}$$

If lev_{ij} is the current level of element a_{ij} then the new level is defined to be

$$\text{lev}_{ij} = \min(\text{lev}_{ij}, \text{lev}_{ik} + \text{lev}_{kj} + 1). \tag{33}$$

The initial level of fill-in for an element a_{ij} of a sparse matrix A is 0 if $a_{ij} \neq 0$ and ∞ otherwise. Each time the element is modified in the GE process its level of fill-in is updated by equation 33. Observe that the level of fill-in of an element will never increase during the elimination. Thus if $a_{ij} \neq 0$ in the original matrix A, then the element will have a level of fill-in equal to zero throughout the elimination process. The above gives a systematic algorithm for discarding elements. Hence ILU(k) contains all of the fill-in elements whose level of fill-in does not exceed k. The algorithm is given by:

```
For all non zero elements $a_{ij}$ define  lev $a_{ij} = 0$
For $i = 2, \ldots, n$ Do:
        For $j = 1, 2, \ldots, i-1$ and for  lev $a_{ij} < k$
                $a_{ij} = a_{ij}/a_{jj}$
                $a_{il} = a_{il} - a_{ij}a_{jl} \quad l = j+1, \ldots, n$          (34)
                Update the levels of fill in for non zero $a_{ij}$
        EndDo
        If  lev $a_{ij} > k$ then $a_{ij} = 0$
EndDo
```

3.4 Evaluation of Aztec

The starting point for studying the linear solver is the Aztec package which has been used to generate all results to date. As a test case we use the system $A_r x = b$ where

$$A_r = \left[\begin{array}{cc} A & I\omega \\ -I\omega & A \end{array} \right]. \tag{35}$$

Here A is the Jacobian matrix of the CFD equations plus the roll equations. This system is used in the inverse power iterations and is close to that used for the augmented solver.

Various calculations were carried out using a test matrix. First, the value of ω was set to zero to obtain a system with just A. Secondly the problem was solved with a representative value for ω and with the various orderings for ILU and BILU factorisations. Finally, one of these cases was rerun with a second order Jacobian matrix for A. The results are summarised in figure 1.

The following conclusions were drawn for the solution of the linear system with Aztec:

- The augmented linear systems are significantly more difficult to solve than CFD only systems.
- The second order Jacobians cannot be solved with zero fill-in in preconditioners whereas the approximate first order Jacobians can.

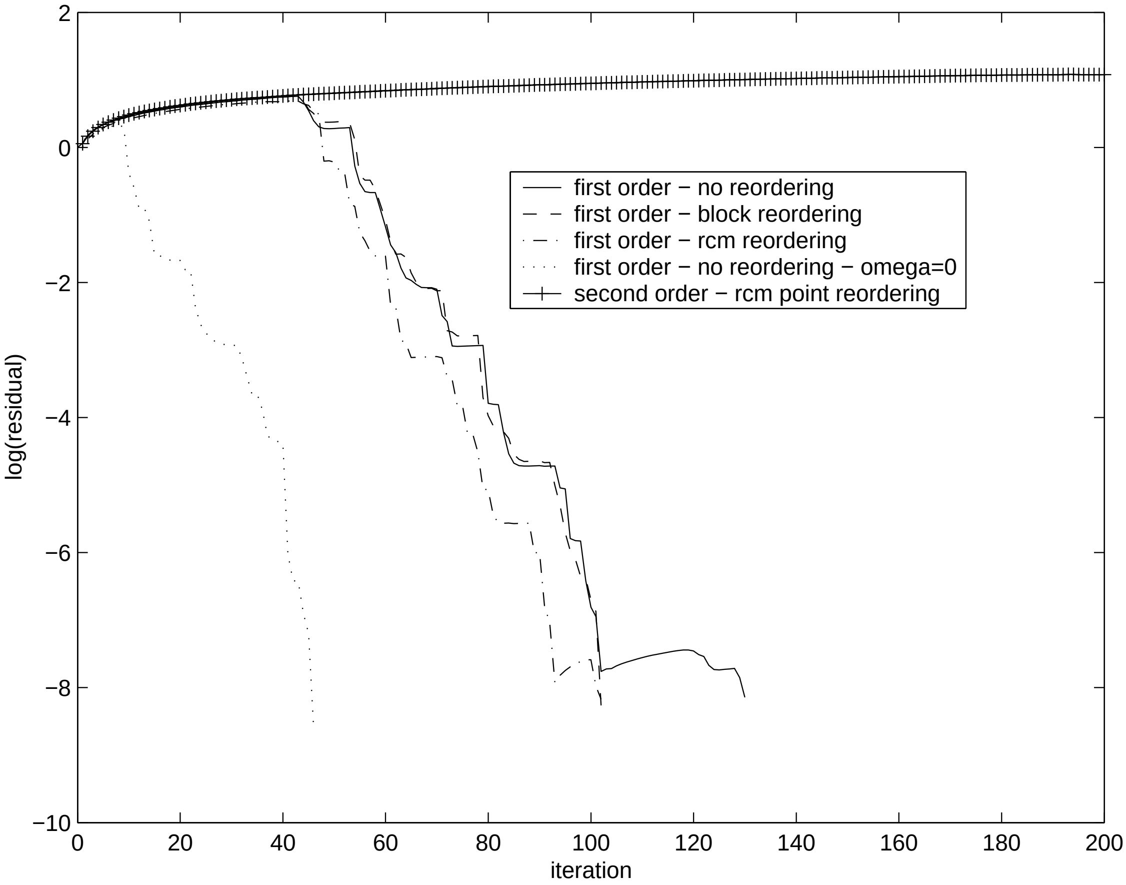

Figure 1: Convergence histories for TFQMR solution of augmented system using several preconditioning options.

To allow an evaluation of the performance of Aztec, and to facilitate easy testing of various options, an implementation of a Krylov method with BILU preconditioning was written in MATLAB. A general version of the preconditioning was written which allows levels of fill-in.
First, tests were carried out using $\omega = 0$. Figure 2 shows the convergence using different preconditioning schemes on the second order Jacobian matrix. Convergence is obtained, even with BILU(0) in contrast to Aztec. BILU(1) gives a rate of convergence approximately equal to the first order Jacobian. Preconditioning the second order Jacobian with a preconditioner from the first order Jacobian works well and saves 45% in storage.
Figure 3 shows the convergence of BILU(0) for the first and second order Jacobains with non zero ω. The effect of adding the non-zero ω terms is minimal.

The conclusion from this initial investigation was that the Aztec package was not performing in an optimal way and that an implementation of the linear solver would have to be coded up. Various options were considered and these are described in the remainder of this section.

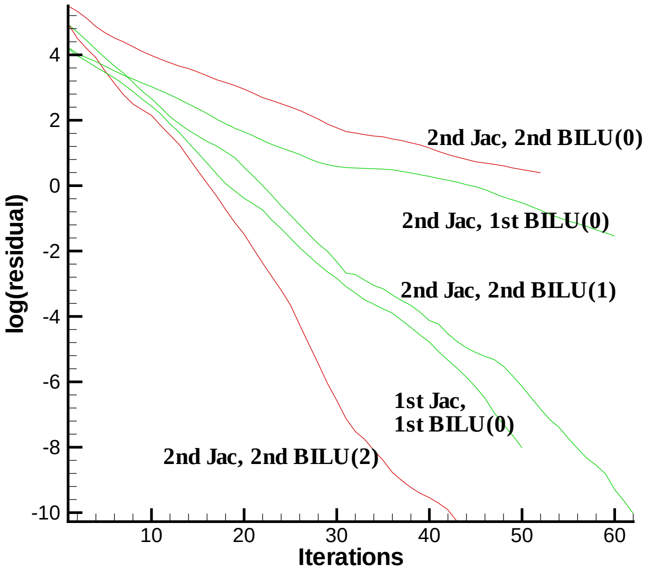

Figure 2: Convergence histories for GCG solution of augmented system with $\omega = 0$ using several preconditioning options.

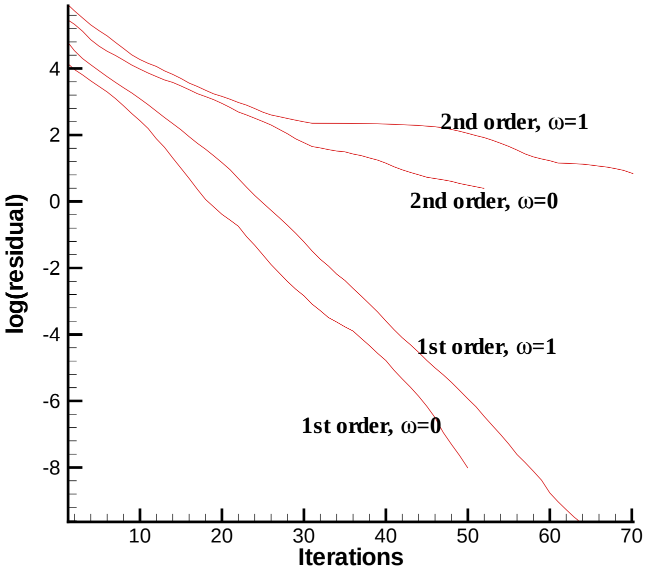

Figure 3: Convergence histories for GCG solution of augmented system with $\omega = 0$ and $\omega = 1.0$ using BILU(0).

3.5 Formulation

The test problem is written in real form above. An alternative is to solve the complex form of the system, written as $(A - i\omega)\mathbf{v} = A_c\mathbf{v} = \mathbf{d}$. The matrix A is further partioned as

$$A = \begin{bmatrix} A_{ff} & A_{fs} \\ A_{sf} & A_{ss} \end{bmatrix}. \tag{36}$$

using the notation introduced in report 2. The real and complex systems then have the following matrices

$$A_c = \begin{bmatrix} A_{ff} & A_{fs} \\ A_{sf} & A_{ss} \end{bmatrix} - i\omega \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix}. \tag{37}$$

$$A_r = \begin{bmatrix} A_{ff} & A_{fs} & \omega I & 0 \\ A_{sf} & A_{ss} & 0 & \omega I \\ -\omega I & 0 & A_{ff} & A_{fs} \\ 0 & -\omega I & A_{sf} & A_{ss} \end{bmatrix}. \tag{38}$$

A number of different preconditioning options for the real and complex systems were tested, using a FORTRAN implementation (which made the treatment of complex variables easier).

Three different methods have been tested from a Incomplete factorization of all of the terms in the matrix to a "blocked" matrix were a large amount of decoupling has been used.

3.5.1 Method 1

This is standard BILU(k) on the unsimplified real A_r or complex matrix A_c.

3.5.2 Method 2

This is BILU(k) on the block diagonal of either A_c or A_r i.e.

$$A_c^b = \begin{bmatrix} A_{ff} & 0 \\ 0 & A_{ss} \end{bmatrix} - i\omega \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} \tag{39}$$

and

$$A_r^b = \begin{bmatrix} A_{ff} & 0 & 0 & 0 \\ 0 & A_{ss} & 0 & 0 \\ 0 & 0 & A_{ff} & 0 \\ 0 & 0 & 0 & A_{ss} \end{bmatrix}. \tag{40}$$

3.5.3 Method 3

This extends the blocking in Method 2 by also including the blocking of the multiblock grid. This means A_{ff} loses all its inter block connectivity following the method used in the Glasgow CFD only solver.

3.6 Results

Table 1 shows the number of non zero 5×5 blocks required for the different methods with a first order Jacobian. Each complex block requires twice the storage of a real block due to the real and imaginary parts. Even taking this into account the complex formulation always uses less memory than the real one. Also the scaling of the memory requirements is increasingly favourable for the complex formulation as the level of fill-in increases. The real formulation of method 2 results in a singular preconditioner due to A_{ss} being singular.

Table 2 shows the number of non zero 5 × 5 blocks required for the different methods with a second order Jacobian. Comparing them with the first order Jacobian with level 3 fill-in in the second order Jacobian requires 4 or 5 times the storage. However there is a much larger decrease in storage requirements as terms in the preconditioner are removed compared to the first order case.

Figure 4 shows the differences between using different size bases in the CGR method for a first order Jacobian and a level zero fill-in. There is sometimes a plateau when the new basis is started due to the simplistic method used to restart - just drop the whole basis and start again.

Figure 5 shows the differences between using different levels of fill-in with the restarted CGR method with a basis of 60 and a first order Jacobian. As the level of fill-in increases the number of iterations to convergence drops. This is mostly due to the reduction in the size of the plateaus after the restart as the gradients outside of these regions are similar.

Figure 6 shows the difference between methods when using level 3 fill-in and 60 bases in the CGR solver with a first order Jacobian. Neglecting A_{fs} and A_{sf} has very little effect on the overall rate of convergence. However removing all the terms that cross the multiblock mesh boundaries cases a large drop off in performance. This has never been seen in the A_{ff} cases and since only two equations have been added to the system it might be be possible to recover the A_{ff} convergence rate.

Figure 7 shows the differences between methods with a basis of 60 in the CGR solver and a second order Jacobian. For the second order Jacobian level 2 fill-in is not adequate for convergence, even for method 1. Level 3 fill-in allows for fast convergence for method 1 and 2 but is slower for method 3. To deal with this we will consider the following. If two blocks are on the same processor then that information will be retained. This means if one processor is used the preconditioner is defined as in method 2 If each processor just has one block the preconditioner is defined as that in method 3. It is possible that the sharp leading edge is causing the problem and hence careful placement of the blocks using this preconditioner may give good performance.

Method	Real or	Number of Non Zeros 5 × 5 Blocks			
	Complex	BILU(0)	BILU(1)	BILU(2)	BILU(3)
1	Real	396518	806558	1590985	N/A
2	Real	N/A as Preconditioner is singular			
1	Complex	175854	304151	511390	902146
2	Complex	151667	278915	485085	874617
3	Complex	141603	247315	402803	689403

Table 1: Table of the number of non zero in the preconditioner for the first order Jacobian

Method	Real or	Number of Non Zeros 5 × 5 Blocks
	Complex	BILU(3)
1	Complex	4273227
2	Complex	4241749
3	Complex	2741219

Table 2: Table of the number of non zero in the preconditioner for the Second order Jacobian

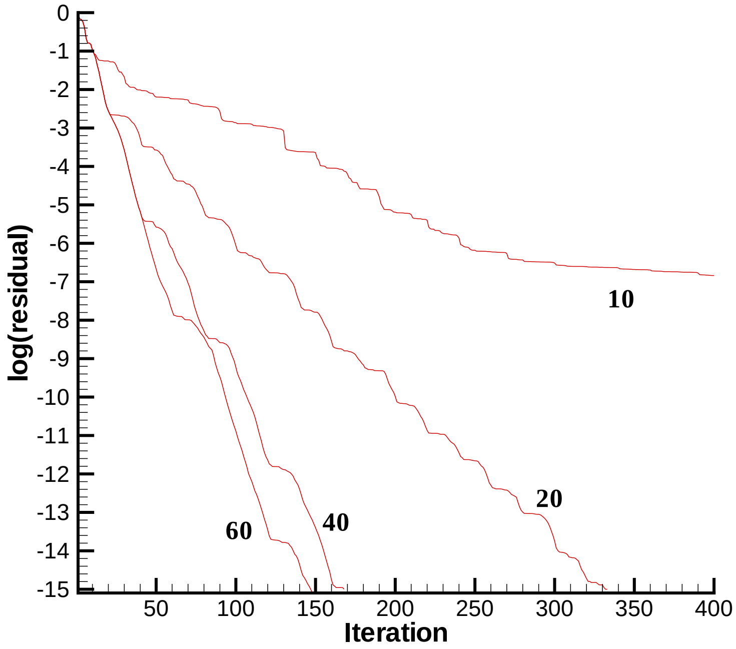

Figure 4: Comparison of the influence of size of the Bases used in the GCR method - first order Jacobian, method 1 preconditioning and no fill-in.

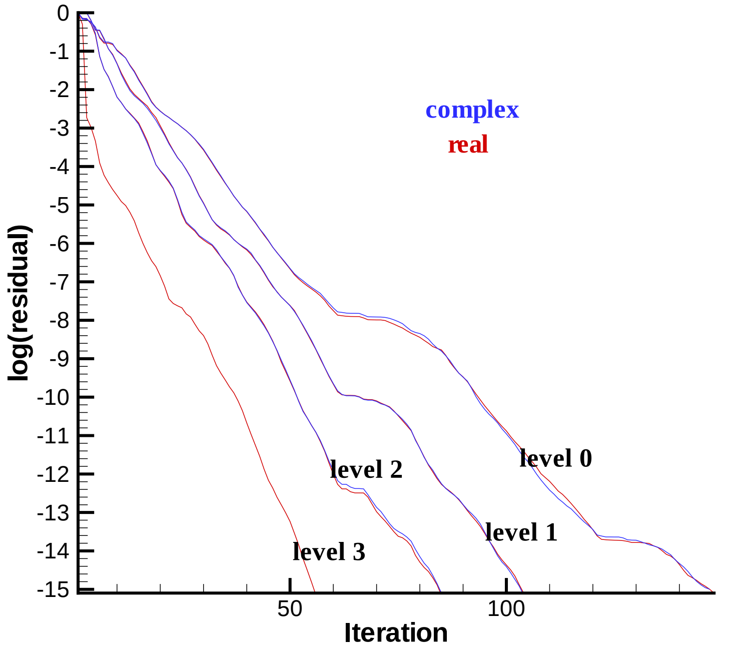

Figure 5: Comparison of the Real and Complex formulations for Method 1 using a first order Jacobian.

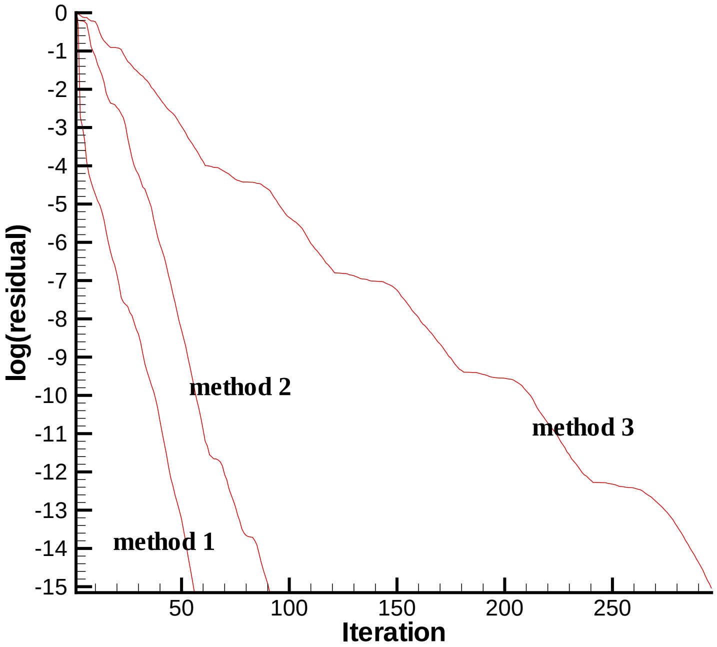

Figure 6: Comparison of the influence of the preconditioning method using the Complex formulation and three levels of fill-in for a first order Jacobian matrix.

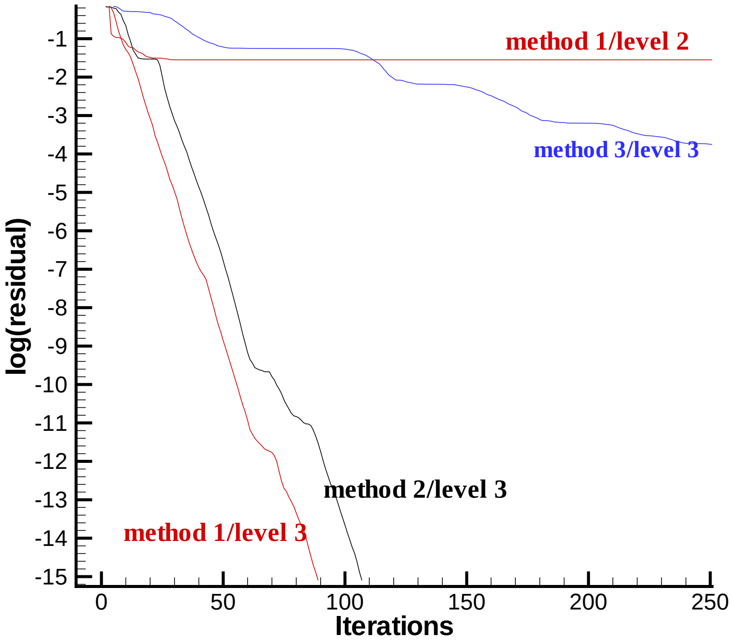

Figure 7: Comparison of the influence of the preconditioning method using the Complex formulation with a second order Jacobian

3.7 Conclusions on Linear Solver

The following conclusions have been drawn

- The complex form is more reliable and uses less storage and hence will be used
- The facility to change the fill-in on the fluid Jacobian is useful
- We can now solve the system using the Jacobian from the second order spatial scheme, the requirement for which was discussed in the introduction
- Method 3 provides a simple way of implementing the preconditioner in parallel
- The FORTRAN implementation is fast compared with the general AZTEC code or the Real C implementation.

4 Parallel Implementation

4.1 Data Decomposition

The data decomposition is done by storing whole blocks of the multiblock grid on just a single processor. This means partitioning of the grid can be consided as a problem of partitioning the blocks. Each processor stores a certain number of blocks in the grid and their associated fluid variables. The structural variables are treated in a different way. Each processor stores all the structural information. This is currently not too expensive as the number of structural equations is small compared to the number of fluid equations, but might have to be revisited for different problems.

4.2 Decomposition of the residual and the Jacobian

The residual and Jacobian are decomposed in the method shown in figure 8. Here 4 processors are assumed and the schematic in the figure indicates

$$\begin{bmatrix} A_{ff} & A_{fs} \\ A_{sf} & A_{ss} \end{bmatrix} \begin{bmatrix} R_f \\ R_s \end{bmatrix}.$$

Processors with no solid boundary conditions will have zeros in the A_{sf} part of the Jacobian.

4.3 Matrix vector and Matrix Transpose vector operations

Since the Jacobian matrix is calculated in 4 distinct parts A_{ff}, A_{fs}, A_{sf} and A_{ss}, the matrix vector product is done in 4 parts which are summed together. The products with A_{fs} and A_{ss} vector are trivial as the structural unknowns are on all processors. The product with A_{sf} requires a “reduce all” at the end of the multiply to get the information to all the processors.

A different ordering is used for the fluid Jacobian of A_{ff}. The cells are partitioned as shown in figure 9, with 3 different sets of cells. The internal set can be updated without any communication and hence it is possible to complete this part of the product while waiting on the messages to arrive from other processors. The border set are the cells owned by this processor but which require some communication with other processors to be fully calculated. The external set is not updated at all on this processor but is used to update the border points. Since the renumbering of A_{ff} has modified both the row and column indices the multiplying vector must be reordered to match. The output product is

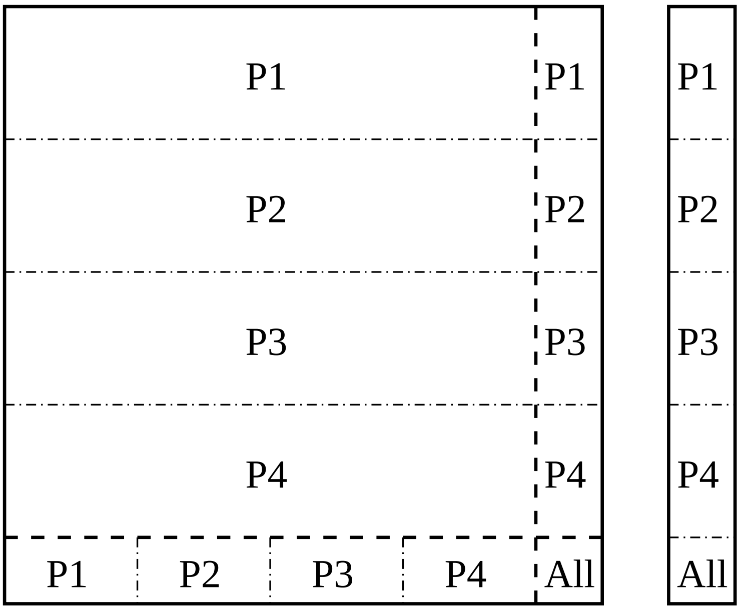

Figure 8: Effective data decomposition of the Jacobian and right hand side vector

also in the reordered state and must be permutated back before doing the preconditioner operations.

The reason why this reordering is only done to the fluid Jacobian is because only the matrix vector product that requires communication to be able to complete. The above renumbering allows for the maximum amount of work to be allocated between sending the halo messages and having to use the halo information from another processor.

As can be seen in figure 10 the matrix transpose vector operation - needed to compute the left eigenvector of the system - looks very similar to the matrix vector operations. The only difference is that for A_{ff} the processors now have complete columns of the fluid variables and not rows. The A_{ff}^T product requires no communication to form the vectors but the external set must be communicated at the end to complete the operation.

4.4 Load Balancing

To maximise the parallel preformance of the code the blocks must be carefully assigned to the processors. If there is a large number of blocks it is not difficult to get approximately the same workload onto each processor for the CFD only case. However in the IPM the workload for non solid wall fluid points is greater than that for any other point - due to the contributions from A_{sf}. Hence the number of fluid points on each processor and the number of solid wall points on each processor should be balanced. Lastly the number of processor block boundaries needs to be kept as small as possible. Not only does this keep the size of the messages smaller but much more importantly BILU(k) can be calculated with no extra appoximations on block boundaries if both blocks are on the same processor.

4.5 Preconditioner

Method 3 is used for implementing the preconditioner in parallel. The preconditioner contains no inter block boundary information and hence each block does not rely on any infomation from any other block. If this preconditioning does not provide convergence it its possible to fill-in the block-block interface information if both blocks are on the same

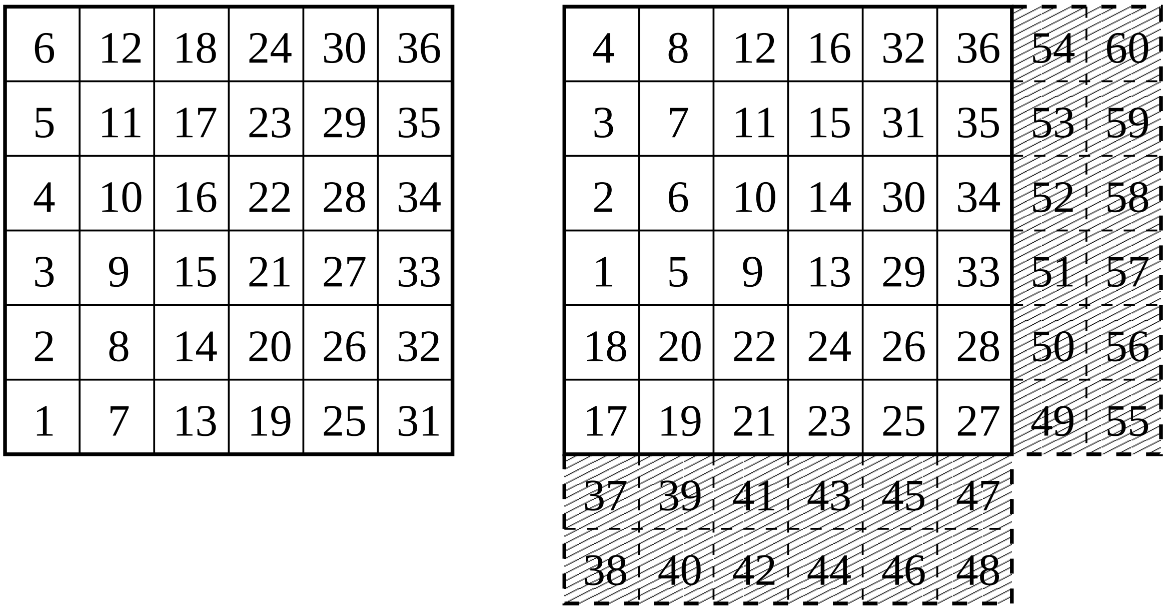

Figure 9: The renumbering scheme used in the Jacobian matrix A_{ff}

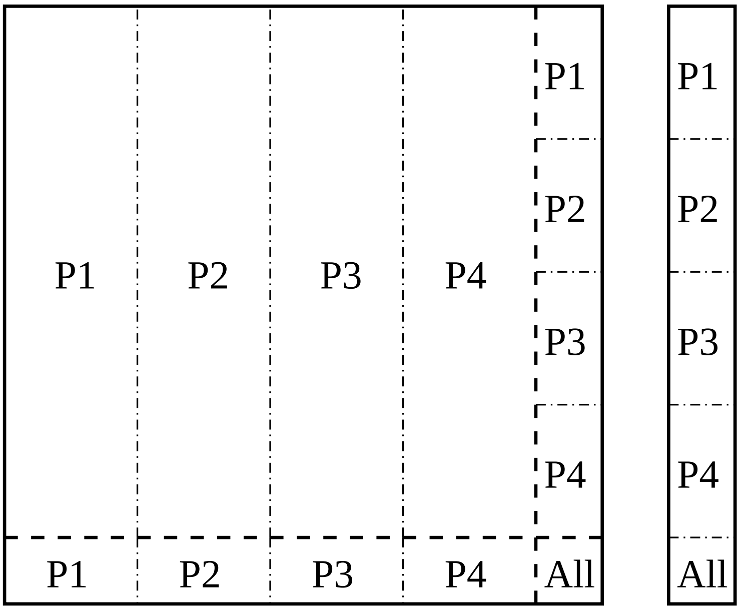

Figure 10: Effective data decomposition of the Jacobian transposed

processor. This however does mean the convergence of the linear system would depend on the partitioning of the blocks to the processors.

4.6 Scaling Results

The results in table 3 were obtained by solving a linear system arising from a second order Jacobian. The speedup on 2 processors is very good considering the small size of the problem. The speedup on 4 processors is lower due to a small 5% load inbalance and a large number of faces needed to be communicated to obtain this load balance. These results are therefore considered very satifactory and open up the capability to tackle fine mesh densities using parallel processing.

Processors	Time in Seconds	Speed up
1	821	
2	419	1.95
4	237	3.46

Table 3: Table of speedups for IPM on a problem with 22,000 cells

5 Test Case and Time Marching Results

5.1 Test Case Description

There is a considerable amount of published experimental data for 80^o sweep delta wings. Hence this sweep angle was selected for the current study. The geometry of the wing was identical to that used by Arena and Nelson [10]. The wing has a flat upper and lower surface with a 45^o windward bevel and a root chord of 0.4222m. The moment of inertia for this wing was given as $I_{xx} = 0.00125$ Kg-m^2. The experiment was performed at a Reynolds number of 1.5×10^5. Since PMB is a compressible flow solver, a Mach number of 0.2 was used to avoid convergence difficulties at the very low Mach numbers used in the experiments. The freestream air density is used to non-dimensionalise the moment of inertia of the wing for the computations. The freestream air density in experiment is unavailable and so the value was assumed to be $1.23Kgm^{-3}$ (sea-level ISA conditions).

5.2 Steady State CFD Tests

Before simulating wing rock, the accuracy of steady solutions was verified with a grid refinement study. A fine grid for Euler simulations was created with approximately 1.6 million points. From this grid two levels were extracted by removing every second grid point in each direction. The RANS solution was performed on a grid with a high resolution in the vortical and boundary layer regions. This grid has 1.8 million points. Steady state solutions were computed for each grid with the residual being reduced 6 orders of magnitude. The test case chosen for validation purposes has the wing at 30^o angle of attack and rolled $+10^o$.

The upper surface pressure distributions from all three Euler grid levels are shown in figure 11 for the chordwise station at $60\%c_r$, and are compared with the measurements of Arena [31]. It should be noted that the experimental results indicated laminar flow on the upper surface of the wing which has the effect of moving the primary vortices inboard

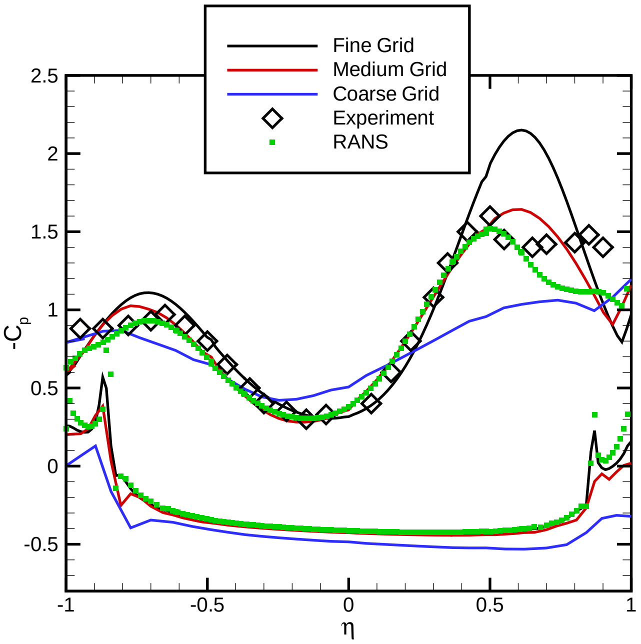

Figure 11: *Grid refinement study - Surface pressure distributions at 60%c_r. Lines indicate the coarse, medium and fine Euler results.*

and upwards off the surface of the wing. This should be kept in mind when considering wing rock which is attributed to hysteretic vortex movement [10]. Examination of the surface pressure distributions from each grid indicates that the solutions are not grid converged. However Euler simulations of delta wing flows are known to be highly sensitive to grid density due to the mechanisms for the generation of vorticity which are based on the truncation error of the discretisation,[32] and therefore the current results are as expected. Despite the requirement to chose an appropriate level of grid density, it is known that Euler simulations can realistically predict the dynamic response of leading edge vortices. Comparing the surface pressure distributions at 60%c_r, there is a reasonably good agreement between the solution from the medium grid and experiment. The effect of viscosity can be seen as a drop in the primary vortex suction peaks, which is due to the presence of a secondary separation shifting the vortex location. As such the RANS suction peaks and locations compare well with experiment.

5.3 Wing Rock Response

The increase in pressure difference between the port and starboard sides as the grid is refined will induce a stronger restoring moment which is likely to produce a stronger wing rock response. This is seen in figure 12 where clearly the largest amplitude wing rock is predicted by the fine grid (followed by the medium and coarse grids). The wing rock amplitudes for the fine, medium, and coarse grids are 70^o, 50^o, and 15^o, with the wing rock amplitude observed in experiment being 40^o.

A comparison of the rolling moments from the medium grid and experiment through one complete steady wing rock cycle is shown in figure 13. Despite the maximum roll angle being exceeded in the Euler solution it is clear that there is reasonably good agreement with experiment, with the rolling moment distribution and magnitudes being predicted well. As discussed by Arena and Nelson [10] there is a clockwise loop where energy is

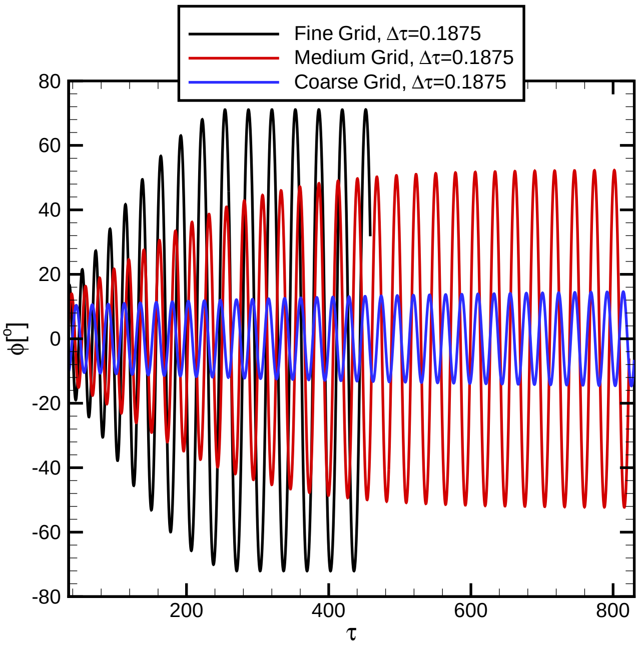

Figure 12: *Grid refinement study - Wing rock histories*

added to the system, and two anti-clockwise damping lobes when energy is dissipated. Comparing the width of the loops where energy is added, more energy is being added to the system in the Euler simulations in comparison to experiment. The damping lobes are also larger in the Euler solution in comparison to experiment.

In order to verify the temporal accuracy of the solutions, a time step refinement study was conducted. It was found that the non-dimensional time step of 0.1875 provides an adequate temporal resolution for the current wing rock studies. This equates to around 120 time steps per cycle. For forced motion studies as few as 50 time steps per cycle are sufficient for temporal convergence, however since in free-to-roll studies errors amplify with time (where auto-rotation was observed to occur when the time step was too large), a smaller time step is required to accurately predict the wing rock behaviour.

Roll angle histories for Euler and RANS simulations of the 80^o sweep delta wing at 30^o angle of attack are shown in figure 14. Comparing first the wing rock amplitudes, the amplitudes from the RANS solution, Euler solution, and experiment are 35^o, 50^o, and 40^o respectively. Clearly the effect of viscosity is to reduce the wing rock amplitude. This is as expected based on the previous discussion. With the higher more outboard suction peaks from the Euler solutions there is a much larger wing rock response in comparison to the RANS solutions, where the suction peaks are lower and more inboard. Comparing the period of the wing rock cycle we can see that the Euler and RANS solutions produce similar results. As such it is concluded that the Euler solutions predict qualitatively the correct vortex dynamics and therefore a realistic wing rock response.

5.4 Onset Angle

Simulations of free-to-roll motion at 15^o and 20^o angle of attack were conducted on the medium grid using the Euler equations. The roll angle histories for both simulations are shown in figure 15. At 15^o angle of attack the solution is dynamically stable (i.e. the amplitude of the oscillations decreases due to aerodynamic damping). However at 20^o angle

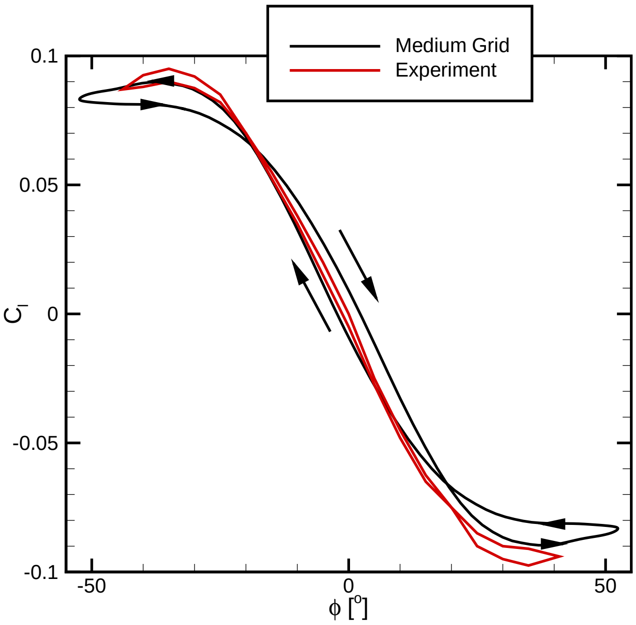

Figure 13: *Comparison of rolling moments through a steady wing rock cycle - CFD and experiment*

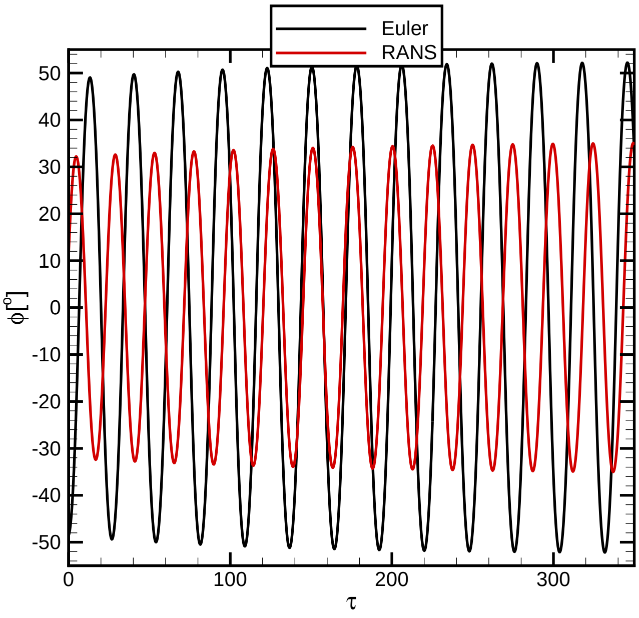

Figure 14: *Effect of modelling - Comparison between RANS and Euler solutions for $\alpha = 30^o$*

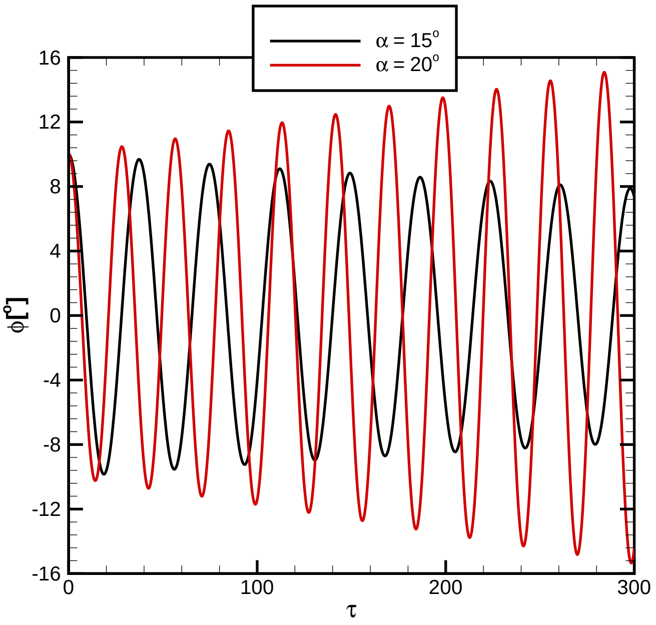

Figure 15: *Wing rock roll angle histories* - $\alpha = 15^o$ *and* 20^o

of attack, the initial roll angle of 10^o initiates a wing rock response with the amplitude of the motion increasing with time. The onset of wing rock observed in experiment was 22^o [10].

6 Fast Methods for Predicting Onset

6.1 Overview

The simulations described above involve the time domain simulation of the wing response to an initial roll perturbation. Multiple time domain calculations to determine the onset angle for the instability are potentially costly. It is therefore of interest to consider methods that retain the full modelling fidelity of the CFD but are computationally more efficient. Various methods were described earlier in the paper; Hopf Bifurcation calculation, inverse power method, linearised solver and proper orthogonal decomposition (POD) reduced order modelling. In this section these methods are applied to the wing rock prediction and their efficiency is evaluated. This is done for the coarse Euler grid, using first and second order spatial discretisations. The results of the previous section suggest that qualitatively the behaviour on this grid is correct, and that methods which can be generalised to solving the RANS equations on fine grids will be of use for quantitative predictions.

6.2 First Order Spatial Results

It is useful to start with a first order spatial discretisation, since for this case $\overline{A} = A$.

The full nonlinear and spatially linearised systems should be able to predict the onset angle through the growth or decay of the response to an initial perturbation. The responses were calculated with a reduced time step of 0.125 at three different angles of

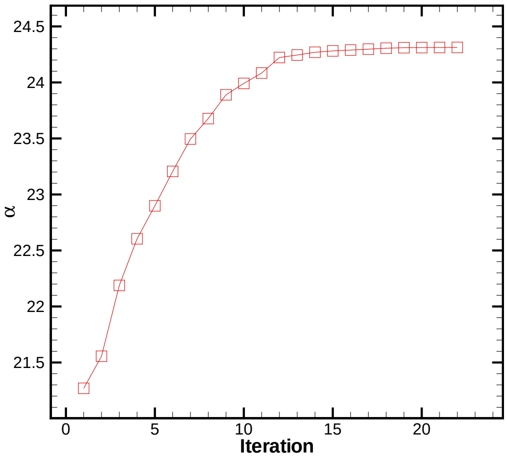

Figure 16: *Convergence of onset angle for the direct calculation (first order).*

attack, namely 20, 24 and 27 degrees. Results for the linearised and nonlinear methods are identical. The response of the roll angle is damped at 20 degrees, grows at 27 degrees and is near to being neutrally stable at 24 degrees. By looking to the ratio of consective peaks in the traces it is possible to estimate that the onset angle is 24.3 degrees.

The inverse power method can be used to calculate the behaviour of the critical eigenvalue as a function of the angle of incidence for the first order spatial scheme. A shift of 0.2 was found to result in convergence to the critical eigenvector. The eigenvalue was found to converge to six decimal places within 7 steps. At 24 degrees the critical eigenvalue is $-1.65 \times 10^{-4} + i0.237$ and at 25 degrees it was $3.46 \times -4 + i0.247$. Linear interpolation suggests that the eigenvalue crosses the imaginary axis at 24.3 degrees.

The direct calculation based on the onset angle was given an initial angle of 20 degrees. From this the first order spatial problem converged rapidly in 20 steps, as shown in figure 16. The converged onset angle is 24.3 degrees. The converged frequency is 0.239, which is also in agreement with the inverse power method and the linearised time domain predictions.

The direct calculation based on the damping coefficient was run at angles of 21, 23 and 25 degrees. The convergence of the onset damping at these angles is shown in figure 17. Linear interpolation between these values suggests that the damping goes to zero at an angle of 24.3 degrees.

Finally, snapshots were collected during one cycle of the linearised time marching calculations. From these POD modes were generated using the matlab script given in appendix A. The linearised system was then projected onto these modes to provide a small order linear system. Six modes were retained for this. The eigenspectrum of the reduced Jacobian at each angle was calculated in matlab. Again, this indicates that at 20 degrees the system is stable, at 24 degrees it is just below neutrally stable and at 27 degrees unstable. By linear interpolation of the real part of the critical eigenvalue it is predicted that the onset angle is 24.4 degrees and the frequency is 0.237.

6.3 Second Order Spatial Results

Next, the second order spatial scheme was used.

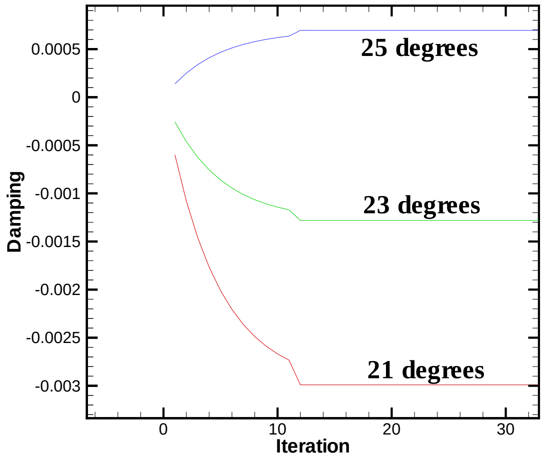

Figure 17: *Convergence of onset damping for the direct calculation (first order).*

Time marching results are shown in figure 18. Again the nonlinear and linearised schemes give identical results. The response at 23 degrees grows whereas the one at 21 degrees decays. Linear interpolation of the amplification gives an onset angle of 22.4 degrees.

The inverse power method was used to calculate the wing rock onset angle. 3 Levels of fill-in were used for the linear solver which allowed the residual of the linear system to be driven down 11 orders in less than 60 iterations.

At 22 degrees, after 7 inverse power iterations the critical eigenvalue had converged to 6 significant figures. The value is $-1.381245 \times 10^{-4} \pm i0.223497$. At 23 degrees the converged value is $1.805895 \times 10^{-4} \pm i0.232095$. These angles bracket the onset angle, which can be estimated by linear interpolation as 22.4 degrees.

The direct calculation to find the onset angle did not convergence. This appeared to be due to the sequencing between the eigenvalue and equilibrium equations. The problem was not present for the first order spatial system, possibly because in that case the eigenvalue equations were hit with a Newton step each time, which resulted in a high level of convergence. The direct calculation based on damping was made at angles of 21 and 23 degrees. The convergence of the damping in each case is shown in figure 19. By linearly interpolating between the damping values predicted the onset angle is estimated to be 22.4 degrees.

Snapshots were again collected during one cycle of the linearised time marching calculations at 21 and 23 degrees. The reduced Jacobian has a critical eigenvalue of $-0.0004 + i0.218$ and $0.00018 + i0.232$ at 21 and 23 degrees respectively.

The cost of the second order calculations is summarised in table 4. The linear and nonlinear time marching results are based on computing five cycles of the response. The reduced model was based on modes computed from one cycle of the linearised solver. The inverse power solver is relatively more costly than the bifurcation solver due to the increased cost of solving the full as opposed to the approximate linear system.

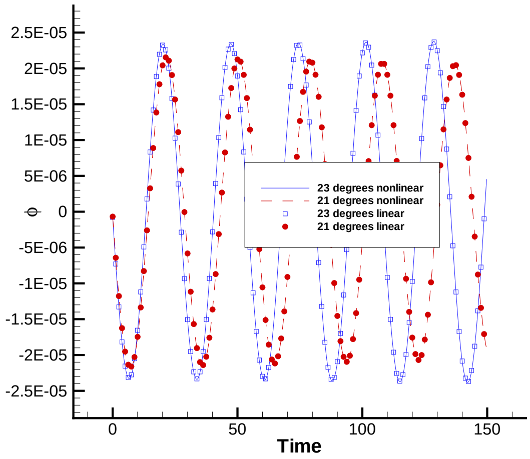

Figure 18: *Response for linear and nonlinear time marching (second order).*

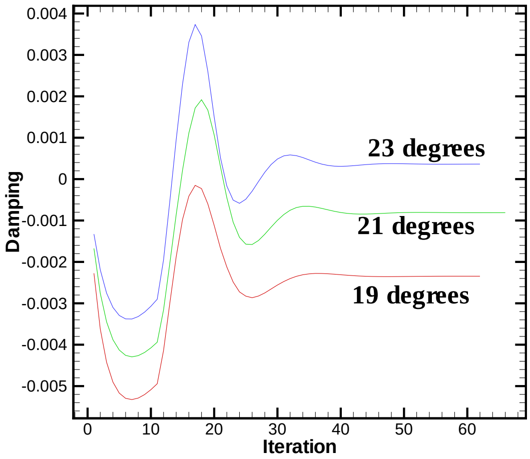

Figure 19: *Convergence of onset damping for the direct calculation with a second order spatial scheme (second order).*

Method	Relative CPU time
steady state	1.00
Bifurcation (Damping)	1.36
Inverse Power	6.00
Linearised Time domain	13.10
Nonlinear Time Domain	81.00
POD reduced model	3.60

Table 4: Calculation times (second order) normalised by the steady state calculation time.

7 Conclusions

A range of methods for predicting wing rock onset have been presented. The basis for the predictions is a residual calculation for the Euler equations. An assessment of the influence of viscous effects, grid dependency and time accuracy was first made. The fast methods considered are direct bifurcation calculation, linearised time domain solution and reduced order modelling based on proper orthogonal decomposition.

The following conclusions have been drawn

- The methods provide consistent results for the onset angle
- The bifurcation calculation for the onset angle was unsatisfactory due to sequencing problems between the mean flow and eigensystem equations.
- An alternative approach based on structural damping proved successful and allowed the onset angle to be calculated efficiently.
- The relative cost of the various methods showed the efficiency of the direct calculation,with the linearised time domain solver being an order of magnitude faster than the nonlinear time domain solver.

8 Acknowledgements

This work has been funded by the European Office of Aerospace Research and Development under contract FA8655-03-1-3044.

A Formulation of Reduced Order Model

The prediction of bifurcation points is only one part of an analysis which is likely to be required. The pre and post bifurcation behaviour is probably of much more interest. For example, flight tests monitor damping characteristics and for a theory to be of use in reducing risk and cost, the pre-bifurcation damping must be readily available. There is also much interest in limit cycle behaviour post bifurcation.

This implies that a fast method to predict the bifurcation point is not enough. In addition, methods are required to predict behaviour for parameter values in the proximity of the bifurcation point. If these methods revert to the full order system of equations then much of the benefit of the fast onset prediction will be lost. However, the structure of the dynamical system close to the bifurcation point can be exploited to produce small order models which can predict the important dynamics and which are computationally efficient.

In this section the formulation of two approaches to be examined is given, along with background information.

A.1 Centre Manifold Reduction

Consider the nonlinear system of equations

$$\dot{\mathbf{w}} = \mathbf{R}(\mathbf{w}, \alpha), \quad \mathbf{w} \in \Re^n, \quad \alpha \in \Re^m \tag{41}$$

where $\mathbf{R}$ is sufficiently smooth. We assume that we are at a Hopf bifurcation at $\alpha = 0$ and hence the Jacobian matrix $\partial \mathbf{R}/\partial \mathbf{w}$ has 2 and only 2 critical eigenvalues with zero real part and the remaining $n_s = n - 2$ eigenvalues have negative real parts. Then the system (41) can be transformed to

$$\left\{ \begin{array}{lcl} \dot{\mathbf{w}}_c &=& A_c(\alpha)\mathbf{w}_c + \mathbf{g}(\mathbf{w}_c, \mathbf{w}_s, \alpha) \\ \dot{\mathbf{w}}_s &=& A_s(\alpha)\mathbf{w}_s + \mathbf{h}(\mathbf{w}_c, \mathbf{w}_s, \alpha) \end{array} \right. \tag{42}$$

where $\mathbf{w}_c \in \Re^2$ and $\mathbf{w}_s \in \Re^{n_c}$. $A_c(0)$ is a 2×2 matrix with its eigenvalues on the imaginary axis and $A_s(0)$ is a $n_c \times n_c$ matrix with no eigenvalues on the imaginary axis. The functions $\mathbf{g}$ and $\mathbf{h}$ have at least quadratic terms. The system 42 has a family of invariant manifolds, locally representable for small $||\alpha||$ as

$$W^c_\alpha = \{(\mathbf{w}_c, h^\alpha(\mathbf{w}_c, \alpha)) \quad : \mathbf{w}_c \in \Re^2, \quad ||\mathbf{w}_c|| \leq \epsilon\} \tag{43}$$

where $\epsilon > 0$ is sufficiently small and $h^\alpha : \Re^2 \times \Re^m \to \Re^{n_s}$ is smooth. Moreover, $h^\alpha(\mathbf{w}_c, 0) = h^c(\mathbf{w}_c)$, i.e. W^c_0 coinsides with the centre manifold of W^c the system with no paramater variation.

The Reduction Principle says system (42) is locally topologically equivalent near the origin to

$$\left\{ \begin{array}{lcl} \dot{\mathbf{w}}_c &=& A_c(\alpha)\mathbf{w}_c + \mathbf{g}(\mathbf{w}_c, h^\alpha(\mathbf{w}_c, \alpha), \alpha) \\ \dot{\mathbf{w}}_s &=& A_s(0)\mathbf{w}_s \end{array} \right. \tag{44}$$

The important thing to notice is that the equations for $\mathbf{w}_c$ and $\mathbf{w}_s$ are decoupled in equation (44). The first equation is the restriction of equation (42) to its centre manifold. The dynamics of the structurally unstable system (42) are essentially determined by this restriction, since the second equation in (44) is linear. For a Hopf bifurcation with ($\lambda_{1,2} = \pm i\omega$) then the system looks like

$$\left\{ \begin{array}{lcl} \begin{pmatrix} \dot{w_{c1}} \\ \dot{w_{c2}} \end{pmatrix} &=& \begin{pmatrix} 0 & -\omega \\ \omega & 0 \end{pmatrix} \begin{pmatrix} w_{c1} \\ w_{c2} \end{pmatrix} + \begin{pmatrix} G_1(w_{c1}, w_{c2}, \mathbf{w}_s, \alpha) \\ G_2(w_{c1}, w_{c2}, \mathbf{w}_s, \alpha) \end{pmatrix} \\ \dot{\mathbf{w}}_s &=& A_s\mathbf{w}_s + H(w_{c1}, w_{c2}, \mathbf{w}_s, \alpha) \end{array} \right. \tag{45}$$

Due to the tangent property of W^c_0 we can parameterise W^c_α for small $|\alpha|$ via a means of a projection of W^c_α onto $T^c = \{\mathbf{w}_s = 0\}$. For a first approximation we shall assume α has no effect on this parameterization.

It is possible to rewrite this in complex form by use of the variable $z = w_{c1} + iw_{c2}$ to obtain

$$\left\{ \begin{array}{lcl} \dot{z} &=& i\omega z + G(z, \bar{z}, \mathbf{w}_s) \\ \dot{\mathbf{w}}_s &=& A_s\mathbf{w}_s + H(z, \bar{z}, \mathbf{w}_s) \end{array} \right. \tag{46}$$

where G and H are smooth complex-valued functions of z, $\bar{z} \in C^1$ of at least quadratic order. The centre manifold W^c can be locally represented as a graph of a smooth function

$$W^c = \{(z, \mathbf{w}_s) : \mathbf{w}_s = h(z)\}$$

h maps $\Re^2 \to \Re^{n-2}$ and due to the tangent property of W^c, $h(z) = O||\mathbf{w}_c||^2$. The Centre manifold W^c therefore has the representation

$$\mathbf{w}_s = h(z, \bar{z}) + \frac{1}{2}w_{20}z^2 + w_{11}z\bar{z} + \frac{1}{2}w_{02}\bar{z}^2 + O(|z|^3), \tag{47}$$

with the coefficients $w_{ij} \in C^2$. Since $\mathbf{w}_s$ must be real, w_{11} is real and $w_{20} = \bar{w}_{02}$. Using Taylor expansions in z, $\bar{z}$, and $\mathbf{w}_s$ the system (46) can be rewritten as

$$\left\{ \begin{array}{lcl} \dot{z} &=& i\omega z + \frac{1}{2}G_{20}z^2 + G_{11}z\bar{z} + \frac{1}{2}G_{02}\bar{z}^2 \\ &+& \frac{1}{2}G_{21}z^2\bar{z} + \langle G_{10}, \mathbf{w}_s\rangle z + \langle G_{01}, \mathbf{w}_s\rangle \bar{z} + \ldots \\ \dot{\mathbf{w}}_s &=& A_s\mathbf{w}_s + \frac{1}{2}H_{20}z^2 + H_{11}z\bar{z} + \frac{1}{2}H_{02}\bar{z}^2 + \ldots \end{array} \right. \tag{48}$$

where G_{20}, G_{11}, G_{02}, $G_{21} \in C^1$ and G_{01}, G_{10}, $H_{ij} \in C^{n-2}$. Since $\mathbf{w}_s$ is real H_{11} is real, and $H_{20} = \bar{H}_{02}$.

$$G_{jk} = \left.\frac{\partial^{j+k}}{\partial z^j \partial \bar{z}^k} G(z, \bar{z}, 0)\right|_{z=0}, \quad j + k \geq 2, \tag{49}$$

$$\bar{G}_{10,j} = \left.\frac{\partial^2}{\partial \mathbf{w}_{sj} \partial z} G(z, \bar{z}, \mathbf{w}_s)\right|_{z=0, \mathbf{w}_s=0}, \quad j = 1, 2, \ldots, n-2, \tag{50}$$

$$\bar{G}_{01,j} = \left.\frac{\partial^2}{\partial \mathbf{w}_{sj} \partial \bar{z}} G(z, \bar{z}, \mathbf{w}_s)\right|_{z=0, \mathbf{w}_s=0}, \quad j = 1, 2, \ldots, n-2, \tag{51}$$

$$H_{jk} = \left.\frac{\partial^{j+k}}{\partial z^j \partial \bar{z}^k} H(z, \bar{z}, 0)\right|_{z=0}, \quad j + k = 2, \tag{52}$$

On substituting the representation of the centre manifold in (48) and equating coeffients,

$$\begin{array}{rcl} w_{20} & = & (2i\omega I - A_s)^{-1} H_{20} \\ w_{11} & = & -C^{-1} H_{11} \\ w_{02} & = & (-2i\omega I - A_s)^{-1} H_{20} \end{array} \tag{53}$$

Where I is the identity matrix and the matrices $(2i\omega I - A_s)$, A_s and $(-2i\omega I - A_s)$ are invertible since 0 and $\pm 2i\omega$ are not eigenvalues of A_s.

A.2 Projection Based Reduction

The centre manifold theory in principle can be used to reduce the system to two degrees of freedom in the vicinity of the bifurcation point. However, there are considerable difficulties of implementation when using systems of large degree and functions of the complexity encountered in the wing rock project. To overcome this a method is proposed whose main objective is to maintain the original structure of the problem as much as possible, allowing Jacobian matrices of the original system to be used as much as possible.

The full system can be transformed by using only the vectors corresponding to the critical eigenvalues of A and its transpose A^T. These are calculated by using the direct method for the onset angle. The system is projected onto its critical eigenspace and complement. Suppose we have a Taylor expansion of the residual function $\mathbf{R}$ about the equilibrium solution $\mathbf{w}_0$ and parameter at the bifurcation point μ_0, giving

$$\dot{\hat{\mathbf{w}}} = A\hat{\mathbf{w}} + \mathbf{F}(\hat{\mathbf{w}}, \hat{\mu}), \quad \mathbf{w} \in \Re^n \tag{54}$$

where $\mathbf{F}(\hat{\mathbf{w}}, \hat{\mu})$ has at least quadratic terms and $\hat{\mathbf{w}} = \mathbf{w} - \mathbf{w}_0$, $\hat{\mu} = \mu - \mu_0$. The matrix A has a pair of complex eigenvalues on the imaginary axis $\lambda_{1,2} = i\omega$, $\omega > 0$. Let $\mathbf{q}$ be the right eigenvector corresponding to λ_1. Then $\bar{\mathbf{q}}$ is the right eigenvector corresponding to λ_2 and

$$A\mathbf{q} = i\omega\mathbf{q}, \quad A\bar{\mathbf{q}} = -i\omega\bar{\mathbf{q}}$$

The left eigenvector $\mathbf{p}$ has the same property

$$A^T\mathbf{p} = -i\omega\mathbf{p}, \quad A^T\bar{\mathbf{p}} = i\omega\bar{\mathbf{p}}.$$

These can be normalised such that $\langle \mathbf{p}, \mathbf{q} \rangle = 1$ where $\langle \mathbf{p}, \mathbf{q} \rangle = \sum_{i=1}^n \bar{p}_i q_i$. The eigenspace S corresponding to $\pm i\omega$ is two dimensional and is spanned by $\{\Re\mathbf{q}, \Im\mathbf{q}\}$. The eigenspace T corresponds to all the other eigenvalues of A and is $n-2$ dimensional. Then $\mathbf{y} \in T$ if and only if $\langle \mathbf{p}, \mathbf{y} \rangle = 0$. Since $\mathbf{y} \in \Re^n$ while $\mathbf{p}$ is complex then two real constraints on $\mathbf{y}$ exist and hence it is possible to decompose any $\hat{\mathbf{w}} \in \Re^n$ as

$$\hat{\mathbf{w}} = z\mathbf{q} + \bar{z}\bar{\mathbf{q}} + \mathbf{y}$$

where $z \in C^1$, $z\mathbf{q} + \bar{z}\bar{\mathbf{q}} \in S$, and $\mathbf{y} \in T$. The complex variable z is a coordinate of S so

$$\left\{ \begin{array}{lcl} z & = & \langle \mathbf{p}, \hat{\mathbf{w}} \rangle \\ \mathbf{y} & = & \hat{\mathbf{w}} - \langle \mathbf{p}, \hat{\mathbf{w}} \rangle \mathbf{q} - \langle \bar{\mathbf{p}}, \hat{\mathbf{w}} \rangle \bar{\mathbf{q}} \end{array} \right.$$

since $\langle \mathbf{p}, \bar{\mathbf{q}} \rangle = 0$. So the equation (54) has the form

$$\left\{ \begin{array}{lcl} \dot{z} & = & i\omega z + \langle \mathbf{p}, F(z\mathbf{q} + \bar{z}\bar{q} + \mathbf{y}, \hat{\mu}) \rangle \\ \dot{\mathbf{y}} & = & A\mathbf{y} + F(z\mathbf{q} + \bar{z}\bar{\mathbf{q}} + \mathbf{y}, \hat{\mu}) - \langle \mathbf{p}, F(z\mathbf{q} + \bar{z}\bar{\mathbf{q}} + \mathbf{y}, \hat{\mu}) \rangle \mathbf{q} - \langle \mathbf{p}, F(z\mathbf{q} + \bar{z}\bar{\mathbf{q}} + \mathbf{y}, \hat{\mu}) \rangle \bar{\mathbf{q}} \end{array} \right.$$

This system is $(n + 2)$ dimensional but we have two constraints on $\mathbf{y}$.

A.2.1 Simplified Damping Model

Now, in general at this stage a centre manifold reduction should be used to obtain a relationship between $\mathbf{y}$ and z which allows the critical dynamics to be calculated from the z equation only. This treatment allows nonlinear features such as Limit Cycle Oscillations to be calculated from the reduced model. A description of the method to perform this reduction is given in appendix A, and is referred to in the results section as the *centre manifold reduced model.* Here we are interested in calculating damping for parameter values below the bifurcation point. In this case the influence of the component $\mathbf{y}$ from the non-critical space is damped faster than the critical component z. We therefore neglect the influence of $\mathbf{y}$ altogether which removes the need for the central manifold reduction. Further justification for this approximation will be given from results for a test problem below.

Now, in general at this stage a centre manifold reduction should be used to obtain a relationship between $\mathbf{y}$ and z which allows the critical dynamics to be calculated from the z equation only. This treatment allows nonlinear features such as Limit Cycle Oscillations to be calculated from the reduced model. A description of the method to perform this reduction is given in appendix A, and is referred to in the results section as the *centre manifold reduced model.* Here we are interested in calculating damping for parameter values below the bifurcation point. In this case the influence of the component $\mathbf{y}$ from the non-critical space is damped faster than the critical component z. We therefore neglect the influence of $\mathbf{y}$ altogether which removes the need for the central manifold reduction. Further justification for this approximation will be given from results for a test problem below.

The damping is therefore determined by solving the equation

$$\dot{z} \quad = \quad i\omega z + \langle \mathbf{p}, \mathbf{F}(z\mathbf{q} + \bar{z}\bar{\mathbf{q}}, \hat{\mu}) \rangle \ .$$

This system is two dimensional. Finally, we need to calculate the form of $\mathbf{F}$. Expanding the function $\mathbf{R}$ in a Taylor series about the equilibrium solution $\mathbf{w}_0$ and parameter μ_0, noting that in fact the bifurcation parameter is the angle of attack α, gives

$$\begin{array}{lcl} \mathbf{R}(\mathbf{w}, \alpha) & = & \mathbf{R}(\mathbf{w}_0, \alpha_0) + A\hat{\mathbf{w}} + B(\hat{\mathbf{w}}, \hat{\mathbf{w}}) + C(\hat{\mathbf{w}}, \hat{\mathbf{w}}, \hat{\mathbf{w}}) + \dfrac{\partial \mathbf{R}}{\partial \alpha}\hat{\alpha} \qquad (55) \\ & & \dfrac{1}{2}\dfrac{\partial^2 \mathbf{R}}{\partial \alpha^2}\hat{\alpha}^2 + A_\alpha \hat{\alpha} + \dfrac{1}{2}A_{\alpha\alpha}\hat{\alpha}^2\hat{\mathbf{w}} + \dfrac{1}{2}\dfrac{\partial B(\hat{\mathbf{w}}, \hat{\mathbf{w}})}{\partial \alpha}\hat{\alpha}. \qquad (56) \end{array}$$

A.2.2 Full Reduction

The full reduction requires the application of the centre manifold theory to bring in the influence of the components $\mathbf{y}$ on the critical coordinates in the two-degree-of-freedom model.

This system is Taylor expanded in z, $\bar{z}$ and y to give the following approximation

$$\left\{\begin{array}{lcl} \dot{z} & = & i\omega z + \frac{1}{2}G_{20}z^2 + G_{11}z\bar{z} + \frac{1}{2}G_{02}\bar{z}^2 + \frac{1}{2}G_{21}z^2\bar{z} + \langle G_{10}, \mathbf{y}\rangle z + \langle G_{01}, \mathbf{y}\rangle \bar{z} + \ldots \\ \dot{\mathbf{y}} & = & A\mathbf{y} + \frac{1}{2}H_{20}z^2 + H_{11}z\bar{z} + H_{02}\bar{z}^2 + \ldots \end{array}\right. \tag{57}$$

where G_{20}, G_{11}, G_{02}, $G_{21} \in C^1$; G_{01}, G_{10}, $H_{ij} \in C^n$. This approach leads to the scalars and vectors all being functions of $\mathbf{F}$ or inner products of $\mathbf{p}$ and $\mathbf{F}$ (i.e. the original function). The manipulation of this system is feasible, even for systems of large dimension. The centre manifold can be represented by

$$\mathbf{y} = h(z, \bar{z}) = \frac{1}{2}h_{20}z^2 + h_{11}z\bar{z} + \frac{1}{2}h_{02}\bar{z}^2 + \mathcal{O}|z|^3$$

with the constraint $\langle \mathbf{p}, h_{ij}\rangle = 0$. The vectors $h_{ij} \in C^n$ can be found from the linear equations

$$\left\{\begin{array}{rcl} (2i\omega I - A)h_{20} & = & H_{20} \\ -Ah_{11} & = & H_{11} \\ (-2i\omega I - A)h_{02} & = & H_{02} \end{array}\right. \tag{58}$$

These equations are invertable since 0, and $\pm 2i\omega$ are not eigenvalues of A. We can now write the restricted equation as

$$\begin{array}{rcl} \dot{z} & = & i\omega z + \frac{1}{2}G_{20}z^2 + G_{11}z\bar{z} + \frac{1}{2}G_{02}\bar{z}^2 \\ & + & \frac{1}{2}(G_{21} - 2\langle G_{10}, A^{-1}H_{11}\rangle + \langle G_{01}, (2i\omega I - A)^{-1}H_{20}\rangle)z^2\bar{z} + \ldots \end{array}$$

If we write $\mathbf{F}(\mathbf{w})$ in terms of functions $B(x, y)$ and $C(x, y, z)$ defined as

$$B(\mathbf{w}_1, \mathbf{w}_2) = \frac{1}{2}\frac{\partial^2 R}{\partial w^2}\mathbf{w}_1\mathbf{w}_2$$

$$C(\mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3) = \frac{1}{6}\frac{\partial^3 R}{\partial w^3}\mathbf{w}_1\mathbf{w}_2\mathbf{w}_3$$

then

$$\mathbf{F}(\mathbf{w}) = B(\mathbf{w}, \mathbf{w}) + C(\mathbf{w}, \mathbf{w}, \mathbf{w}) + \mathcal{O}\|\mathbf{w}\|^4.$$

Then we can express

$$\langle G_{10}, \mathbf{y}\rangle = \langle \mathbf{p}, B(\mathbf{q}, \mathbf{y})\rangle, \qquad \langle G_{01}, \mathbf{y}\rangle = \langle \mathbf{p}, B(\bar{\mathbf{q}}, \mathbf{y})\rangle$$

and hence the restricted equation is in the form

$$\begin{array}{rcl} \dot{z} & = & i\omega z + \frac{1}{2}G_{20}z^2 + G_{11}z\bar{z} + \frac{1}{2}G_{02}\bar{z}^2 \\ & + & \frac{1}{2}(G_{21} - 2\langle \mathbf{p}, B(\mathbf{q}, A^{-1}H_{11})\rangle + \langle \mathbf{p}, B(\bar{\mathbf{q}}, (2i\omega I - A)^{-1}H_{20})\rangle)z^2\bar{z} + \ldots \end{array}$$

where

$$G_{20} = \langle \mathbf{p}, B(\mathbf{q}, \mathbf{q})\rangle \quad G_{11} = \langle \mathbf{p}, B(\mathbf{q}, \bar{\mathbf{q}})\rangle \quad G_{02} = \langle \mathbf{p}, B(\bar{\mathbf{q}}, \bar{\mathbf{q}})\rangle \quad G_{21} = \langle \mathbf{p}, C(\mathbf{q}, \mathbf{q}, \bar{\mathbf{q}})\rangle$$

and

$$\left\{\begin{array}{l} H_{20} = B(\mathbf{q}, \mathbf{q}) - \langle \mathbf{p}, B(\mathbf{q}, \mathbf{q})\rangle \mathbf{q} - \langle \bar{\mathbf{p}}, B(\mathbf{q}, \mathbf{q})\rangle \bar{\mathbf{q}} \\ H_{11} = B(\mathbf{q}, \bar{\mathbf{q}}) - \langle \mathbf{p}, B(\mathbf{q}, \bar{\mathbf{q}})\rangle \mathbf{q} - \langle \bar{\mathbf{p}}, B(\mathbf{q}, \bar{\mathbf{q}})\rangle \bar{\mathbf{q}} \end{array}\right.$$

Using these equations and the identities

$$A^{-1}\mathbf{q} = \frac{1}{i\omega}\mathbf{q} \quad A^{-1}\bar{\mathbf{q}} = \frac{1}{i\omega}\bar{\mathbf{q}} \quad (2i\omega I - A)^{-1}\mathbf{q} = \frac{1}{i\omega}\mathbf{q} \quad (2i\omega I - A)^{-1}\bar{\mathbf{q}} = \frac{1}{3i\omega}\bar{\mathbf{q}}$$

the restricted equation can be rewritten as

$$\dot{z} = i\omega z + \frac{1}{2}g_{20}z^2 + g_{11}z\bar{z} + \frac{1}{2}g_{02}\bar{z}^2 + \frac{1}{2}g_{21}z^2\bar{z}$$

where

$$g_{20} = \langle \mathbf{p}, B(\mathbf{q},\mathbf{q})\rangle \quad g_{11} = \langle \mathbf{p}, B(\mathbf{q},\bar{\mathbf{q}})\rangle$$

and

$$\begin{aligned} g_{21} &= \langle \mathbf{p}, C(\mathbf{q},\mathbf{q},\bar{\mathbf{q}})\rangle \\ &- 2\langle \mathbf{p}, B(\mathbf{q}, A^{-1}B(\mathbf{q},\bar{\mathbf{q}}))\rangle + \langle \mathbf{p}, B(\bar{\mathbf{q}}, (2i\omega I - A)^{-1}B(\mathbf{q},\mathbf{q}))\rangle \\ &+ \frac{1}{i\omega}\langle \mathbf{p}, B(\mathbf{q},\mathbf{q})\rangle\langle \mathbf{p}, B(\mathbf{q},\bar{\mathbf{q}})\rangle \\ &- \frac{2}{i\omega}|\langle \mathbf{p}, B(\mathbf{q},\bar{\mathbf{q}})\rangle|^2 - \frac{1}{3i\omega}|\langle \mathbf{p}, B(\bar{\mathbf{q}},\bar{\mathbf{q}})\rangle|^2 \end{aligned}$$

This leads to the following algorithm:

1. calculate μ, P, Q and ω
2. calculate $B(\mathbf{q},\overline{\mathbf{q}})$ and $B(\mathbf{q},\mathbf{q})$
3. solve
$$\begin{cases} (2i\omega I - A)h_{20} &= B(\mathbf{q},\mathbf{q}) \\ Ah_{11} &= B(\mathbf{q},\overline{\mathbf{q}}) \end{cases} \tag{59}$$
4. form $C(\mathbf{q},\mathbf{q},\overline{\mathbf{q}})$
5. form g_{20}, g_{11}, g_{21}
6. time march
$$\dot{z} = i\omega z + \frac{1}{2}g_{20}z^2 + g_{11}z\bar{z} + \frac{1}{2}g_{02}\bar{z}^2 + \frac{1}{2}g_{21}z^2\bar{z}$$

A.3 Perturbation Method

The perturbation method of multiple scales can be applied to determine behaviour close to a bifurcation. This approach was used in [35] for an two dimensional aeroelastic problem and was presented in [36] for the Duffing oscillator and in [37] for a three equation model problem. The idea is use a perturbation parameter ϵ to separate out events at different time scales. The terms of $O(\epsilon)$ provide a description of the linear dynamics of the system, and higher order terms provide information about the effect of nonlinearities. The presentation of the method follows that given in [35].

Write

$$\mathbf{w} = \mathbf{w}_0 + \epsilon\mathbf{w}_1(T_0, T_2) + \epsilon^2\mathbf{w}_2(T_0, T_2) + ... \tag{60}$$

where $T_0 = t$ and $T_2 = \epsilon^2 t$. Then, expanding the residual in a Taylor series about the bifurcating equilibrium $\mathbf{w}_0$, and equating terms in ϵ gives the following relations

$$\mathbf{w}_1 = \Theta\mathbf{p}e^{i\omega T_0} + \overline{\Theta}\overline{\mathbf{p}}e^{-i\omega T_0}$$

$$\mathbf{w}_2 = \Gamma + \overline{\Gamma} \tag{61}$$

$$\dot{\Theta} = \hat{\mu}\beta_1\Theta + 4\beta_2\Theta^2\overline{\Theta} \tag{62}$$

where

$$\Gamma = 2z_0\Theta\overline{\Theta} + 2z_2\Theta^2 e^{2i\omega T_0} \tag{63}$$

$$\beta_1 = \mathbf{q}^T A_\mu \mathbf{p} \tag{64}$$

$$\beta_2 = \mathbf{q}^T(2B(\mathbf{p}, z_0) + B(\overline{\mathbf{p}}, z_2) + \frac{3}{4}C(\mathbf{p}, \mathbf{p}, \overline{\mathbf{p}})) \tag{65}$$

$$Az_0 = -\frac{1}{2}B(\mathbf{p}, \overline{\mathbf{p}}) \tag{66}$$

$$(2i\omega I - A)z_2 = \frac{1}{2}B(\mathbf{p}, \mathbf{p}). \tag{67}$$

The operator A_μ denotes

$$A_\mu = \frac{\partial^2 R}{\partial w \partial \mu}$$

where the Jacobian is evaluated at the bifurcation point. Θ is a complex variable which can be written as $\Theta = \frac{1}{2}ae^{i\theta}$ where (from equation 62)

$$\dot{a} = \hat{\mu}\beta_{1r}a + \beta_{2r}a^3 \tag{68}$$

$$\dot{\theta} = \hat{\mu}\beta_{1i} + \beta_{2i}a^2. \tag{69}$$

This leads to the following algorithm

1. calculate μ, P, Q and ω
2. compute $B(\mathbf{p}, \overline{\mathbf{p}})$ and $B(\mathbf{p}, \mathbf{p})$
3. solve

   $$Az_0 = -\frac{1}{2}B(\mathbf{p}, \overline{\mathbf{p}})$$

   and

   $$(2i\omega I - A)z_2 = \frac{1}{2}B(\mathbf{p}, \mathbf{p})$$

   for z_0 and z_2.
4. compute $B(\mathbf{p}, z_0)$, $B(\overline{\mathbf{p}}, z_2)$ and $C(\mathbf{p}, \mathbf{p}, \overline{\mathbf{p}})$
5. compute $i\mathbf{q}^T(B\mathbf{p})$
6. evaluate β_1 and β_2
7. perturb μ and solve

   $$\dot{a} = \hat{\mu}\beta_{1r}a + \beta_{2r}a^3$$
   $$\dot{\theta} = \hat{\mu}\beta_{1i} + \beta_{2i}a^2.$$

   until convergence
8. calculate $\mathbf{w}$, using the expressions for $\mathbf{w}_1$ and $\mathbf{w}_2$, setting $\epsilon = 1$

B Matlab script for calculating POD modes

```
%% read m mode shapes row wise into u
ut=transpose(u);
C=u*ut;
C=C/m;
[a,lambda] = eigs(C);
phi= ut*a;
phi=phi/sqrt(m*lambda);
%% write out mode shapes from columns of phi
```

C Matlab script for calculating the Jacobian Contributions from MUSCL

```
for (l=0;l<5;l++) {

       d1=c2[l]-c1[l];
       d2=c3[l]-c2[l];
       d3=c4[l]-c3[l];

       den=0.5/(d2*d2+d1*d1+str2);
       lim=(d2*d1+str)*(d2+d1)*den;

       v[l]=c2[l]+lim ;

       vld[0+l*3]=d1*4.0*lim*den - (d2+d1)*d2*den - (d2*d1+str)*den;
       vld[1+l*3]=1.0 - lim*4.0*den*(d1-d2) + (d2-d1)*(d1+d2)*den;
       vld[2+l*3]=-d2*4.0*lim*den + (d2+d1)*d1*den + (d2*d1+str)*den;

       den=0.5/(d2*d2+d3*d3+str2);
       lim=(d2*d3+str)*(d2+d3)*den;

       v[l+5]=c3[l] - lim;

       vld[0+(l+5)*3]=-d2*4.0*lim*den + (d2+d3)*d3*den + (d2*d3+str)*den;
       vld[1+(l+5)*3]=1.0 + lim*4.0*den*(d2-d3) -  (d3-d2)*(d3+d2)*den;
       vld[2+(l+5)*3]=d3*4.0*lim*den - (d2+d3)*d2*den - (d2*d3+str)*den;
 }
```

References

[1] Ross, A. J., “Investigation of nonlinear motion experienced on a slender-wing research aircraft”, J. Aircraft, 99, pp. 625-631, 1971.

[2] Ericsson, L. E., “Wing rock analysis of slender delta wings, review and extension”, J. Aircraft, 32(6), pp. 1221-1226, Nov.-Dec. 1995.

[3] Mabey, D. G., “Similitude relations for buffet and wing rock on delta wings”, Prog. Aerospace Sci., Vol. 33, pp. 481-511, 1997.

[4] Katz, J., “Wing/vortex interactions and wing rock”, Prog. Aerospace Sci., Vol. 35, pp. 727-750, 1999.

[5] Nelson, R. C., and Pelletier, A., “The unsteady aerodynamics of slender wings and aircraft undergoing large amplitude maneuvers”, Prog. Aerospace Sci., Vol. 39, pp. 185-248, 2003.

[6] Matsuno, T. and Nakamura, T., ”Self-induced Roll Oscillation of 45-Degree Delta Wings”, AIAA-2000-0655, 38th AIAA Aerospace Sciences Meeting and Exhibit, 10-13 Jan, 2000, Reno, NM.

[7] Levin, D., and Katz, J., “Self-induced roll oscillations of low-aspect-ratio rectangular wings”, J. Aircraft, 29(4), pp. 698-702, July-Aug. 1992.

[8] Levin, D. and Katz, J., “Dynamic load measurements with delta wings undergoing self-induced roll oscillations”, J. Aircraft, 21 (1), pp. 30-36, Jan. 1984.

[9] Katz, J., and Levin, D., “Self-induced roll oscillations measured on a delta wing / canard configuration”, J. Aircraft, 23(11), pp. 814-819, Nov. 1986.

[10] Arena, A. S. and Nelson, R. C., “Experimental investigations on limit cycle wing rock of slender wings”, J. Aircraft, 31(5), pp. 1148-1155, Sept. - Oct. 1994.

[11] Lee-Rausch, E. M., and Batina, J. T., “Conical Euler analysis and active roll suppression for unsteady vortical flows about rolling delta wings”, NASA Techical Paper 3259, 1993.

[12] Chaderjian, N. M., and Schiff, L. B., “Numerical simulation of forced and free-to-roll delta wing motions”, J. Aircraft, 33(1), pp. 93-99, Jan. - Feb. 1996.

[13] Fritz, W., “WEAG TA 15 Common Exercise V - Numerical and experimental investigation of a delta wing in rolling motion”, DASA Report, DASA-S-R-1820, April 2000.

[14] Soemarwoto, B., Boelens, O., Fritz, W., Allan, M., Ceresola, N., and Buetefisch, K., “Towards the simulation of unsteady maneuvers dominated by vortical flow”, AIAA-2003-3528, June 2003.

[15] Saad, A. A., “Simulation and analysis of wing rock physics for a generic fighter model with three degrees-of-freedom”, PhD Dissertation, Air Force Institute of technology, Wright-Patterson Air Force Base, Dayton, OH, July 2000.

[16] Morton, S.A. and Beran, P.S., Hopf-Bifurcation Analysis of Airfoil Flutter at Transonic Speeds, J Aircraft, 36, pp 421-429, 1999.

[17] Beran, P.S. and Carlson, C.D. Domain-Decomposition Methods for Bifurcation Analysis, AIAA Paper 97-0518, 1997.

[18] Beran, P.S., A Domain-Decomposition Method for Airfoil Flutter Analysis, AIAA Paper 98-0098, 1998.

[19] Badcock, K.J., M.A. Woodgate, M.A. and Richards, B.E., The Application of Sparse Matrix Techniques for the CFD based Aeroelastic Bifurcation Analysis of a Symmetric Aerofoil, AIAA J, 42(5), 883-892, May, 2004.

[20] Badcock, K.J., Richards, B.E., and Woodgate, M.A., "Elements of Computational Fluid Dynamics on block structured grids usin g implicit solvers", Prog. Aerospace Sci., Vol. 36, pp 351-392, 2000.

[21] Badcock, K.J., Woodgate, M.A. and Richards, B.E., Direct Aeroelastic Bifurcation Analysis of a Symmetric Wing Based on the Euler Equations, to appear in Journal of Aircraft, August, 2004.

[22] Jameson, A., "Time dependent calculations using multigrid, with applications to unsteady flows past airfoils and wings", AIAA Paper 91-1596, 1991.

[23] Gordnier, R. E., "Computational study of a turbulent delta-wing flowfield using two-equation turbulence models", AIAA 96-2076, 1996.

[24] Brandsma, F. J., Kok, J. C., Dol, H. S., and Elsenaar, A., "Leading edge vortex flow computations and comparison with DNW-HST wind tunnel data", RTO / AVT Vortex Flow Symposium, Loen, Norway, 2001.

[25] Wilcox, D. C., "Turbulence modelling for CFD", DCW Industries, Inc., La Cañada, California, 1993.

[26] Osher,S. and Chakravarthy, S.R., Upwind Schemes and Boundary Conditions with Applications to Euler equations in General Coordinates , Journal Computational Physics, vol. 50 , 1983, p 447-481.

[27] Eisenstat, S.C., Elman, H.C. and Schultz, M., Variational Iterative Methods for Nonsymmetric Systems of Linear Equations, SIAM Journal of Numerical Analysis, Vol. 20, No. 2, April 1983, 345-357.

[28] Badcock, K.J., Richards, B.E. and Woodgate, M.A., Elements of Computational Fluid Dynamics on Block Structured Grids using Implicit Solvers, Progress in Aerospace Sciences, vol 36, 2000, pp 351-392.

[29] Tuminaro, R.S., Heroux, M., Hutchinson, S.A. and Shahid, J.N., Official Aztec User's Guide Version 2.1, SAND99-8801J, Sandia Laboratory, 1999. Official Aztec User's Guide Version 2.1, SAND99-8801J, Sandia Laboratory, 1999.

[30] Axelsson, O., Iterative Solution Methods, Cambridge University Press, 1994.

[31] Arena, A., "An experimental and computational investigation of slender wings undergoing wing rock", PhD thesis, University of Notre Dame, IN, USA, April 1992.

[32] Allan, M., "A CFD Investigation of wind tunnel interference on delta wing aerodynamics", Ph.D. Thesis, University of Glasgow, Glasgow, UK, October 2002.

[33] Nguyen, L. T., Yip, L., and Chambers, J. R., "Self-induced wing rock of slender delta wings", AIAA Paper 81-1883, Aug. 1981.

[34] Ng, T. T., Malcolm, G. N., and Lewis, L. C., "Flow visualisation study of delta wings in wing-rock motion", AIAA Paper 89-2187, July 1989.

[35] Beran, P.S., Computation of a Limit-Cycle Oscillation using a Direct Method, AIAA Paper.

[36] Nayfeh, A.H. and Balachandran, B., Applied Nonlinear Dynamics, Wiley, New York, 1995.

[37] Nayfeh, A.H. and Balachandran, B., Motion near a Hopf bifurcation of a three-dimensional system, Mech Res Comm, 17, 191-198, 1990.